

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
COMPUTACIÓN**

**Trabajo de titulación previo a la obtención del título de:
Ingeniera e Ingeniero en Ciencias de la Computación**

**TEMA:
DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL MANEJO
DE DATOS EN BODEGA Y ENFERMERÍA DURANTE EL MOVIMIENTO DE
VACUNAS UTILIZADAS EN EL CENTRO DE SALUD TIPO C DEL CENTRO
HISTÓRICO**

**AUTORA Y AUTOR:
WENDY MISHELL GERMAN CAIZA
DORIVAL EMILIO PICHAMBA TUQUERRES**

**TUTOR:
DANIEL GIOVANNY DÍAZ ORTIZ**

Quito, octubre del 2021

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Wendy Mishell German Caiza, con documento de identificación N° 1725674483, y Dorival Emilio Pichamba Tuquerres, con documento de identificación N° 1722380167, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: **DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL MANEJO DE DATOS EN BODEGA Y ENFERMERÍA DURANTE EL MOVIMIENTO DE VACUNAS UTILIZADAS EN EL CENTRO DE SALUD TIPO C DEL CENTRO HISTÓRICO**, mismo que ha sido desarrollado para optar por el título de INGENIERA EN CIENCIAS DE LA COMPUTACIÓN e INGENIERO EN CIENCIAS DE LA COMPUTACIÓN en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Wendy Mishell German Caiza

1725674483

Dorival Emilio Pichamba Tuquerres

1722380167

Quito octubre del 2021

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema:

**DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL MANEJO DE DATOS
EN BODEGA Y ENFERMERÍA DURANTE EL MOVIMIENTO DE VACUNAS
UTILIZADAS EN EL CENTRO DE SALUD TIPO C DEL CENTRO HISTÓRICO** realizado
por Wendy Mishell German Caiza y Dorival Emilio Pichamba Tuquerres, obteniendo un producto que
cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser
considerado como trabajo final de titulación.

Quito, octubre del 2021



Daniel Giovanny Díaz Ortiz

CI: 171697550-1

DEDICATORIA

El presente proyecto está dedicado con todo mi cariño a mi madre, Sansón y Pandora, quienes han sido el pilar y apoyo incondicional durante toda mi formación académica, mi madre me forjó como la persona que soy en la actualidad y muchos de mis logros se lo debo a ella quien se ha encontrado siempre a mi lado motivándome a cumplir mis sueños.

A mis compañeros que han compartido sus conocimientos, alegrías y tristezas durante estos cinco años de la carrera donde nos hemos apoyado logrando que nuestras metas se hagan realidad.

Además, a mi compañera de proyecto por la amistad y apoyo brindado durante todo el proceso de formación universitaria, por animarme a conseguir lo que parecía imposible y cumplir nuestros objetivos de finalizar la carrera.

Dorival Emilio Pichamba Tuquerres

DEDICATORIA

Dedico este proyecto de todo corazón a mi madre, hermanos y hermanas quienes han sido la inspiración para terminar mi carrera profesional, pues son quienes me han forjado, quienes me han enseñado el valor de la familia y quienes han estado en todos mis logros y derrotas, felicitándome o consolándome.

Además, a mi compañero de proyecto por la amistad y apoyo durante toda la carrera académica pues ha demostrado ser un buen compañero y sobre todo un gran amigo.

Wendy Mishell German Caiza

AGRADECIMIENTO

Agradecemos a Dios por la vida, la salud y por guiarnos en nuestro camino impulsando nuestra fuerza de voluntad para llegar a cumplir varios objetivos.

A nuestra familia por el apoyo incondicional en nuestra formación tanto personal como profesional motivándonos a lidiar con obstáculos complejos y cumplir nuestros sueños.

A la Universidad Politécnica Salesiana por brindarnos la oportunidad de continuar nuestra formación profesional.

Al ingeniero Daniel Giovanny Díaz Ortiz por habernos guiado amablemente como tutor en el desarrollo del proyecto de titulación.

Al Msc. Richard Murillo por coordinar y apoyar desde el Ministerio de Salud Pública en todo el proceso de desarrollo del proyecto de titulación.

Al estadista Giovanni Andino, la Lic. Lucía Gutierrez, la Lic. Anita Carpio y su grupo de trabajo porque con sus guías y retroalimentación fueron posibles para realizar el presente proyecto.

Finalmente, agradecemos a todos los docentes de la Universidad Politécnica Salesiana que han sido parte de nuestra formación académica y nos han fomentado conocimiento sobre nuestra carrera y la vida profesional.

Wendy Mishell German Caiza

Dorival Emilio Pichamba Tuquerres

ÍNDICE

CESIÓN DE DERECHOS DE AUTOR.....	2
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR	3
DEDICATORIA	4
DEDICATORIA	5
AGRADECIMIENTO	6
RESUMEN	12
ABSTRACT.....	13
INTRODUCCIÓN	1
Antecedentes.....	1
Justificación	2
Objetivos.....	3
Objetivo General.....	3
Objetivos Específicos.....	3
Marco Metodológico.....	3
CAPÍTULO 1 ESTADO DEL ARTE	5
1.1 Marco Institucional	5
1.2 Marco Teórico	6
1.2.1 Sistema Informático	6
1.2.2 Metodología Scrum.....	7
1.2.3 Herramientas de Desarrollo	8
CAPÍTULO 2 ANÁLISIS Y DISEÑO	11

2.1	Análisis de Requerimientos	11
2.1.1	Requerimientos Funcionales	11
2.1.2	Requerimientos no Funcionales	12
2.2	Historias de Usuario	12
2.3	Diagramas de Casos de Uso	19
2.4	Diagramas de Secuencia	24
2.5	Diagrama Conceptual de la Base de Datos	32
CAPÍTULO 3 CONSTRUCCIÓN Y PRUEBAS		38
3.1	Arquitectura Propuesta	38
3.2	Diagrama de Despliegue	38
3.3	Diagrama Físico de la Base de Datos	40
3.4	Métodos y Algoritmos Generados	41
3.5	Pruebas	55
3.5.1	Caja Negra	55
3.5.2	Carga	60
3.5.3	Stress	64
3.6	Diagnóstico de Resultados	65
Conclusiones		67
Recomendaciones		68
Referencias		69
Anexos		70

INDÍCE DE TABLAS

Tabla 1 Comparación de Metodologías SCRUM, XP, KANBAN	4
Tabla 2 Requerimientos Funcionales	11
Tabla 3 Requerimientos no Funcionales	12
Tabla 4 Inicio de Sesión	12
Tabla 5 Crear Usuarios y Asignar Roles	13
Tabla 6 Registrar vacunas de los establecimientos	14
Tabla 7 Visualizar inventario de las vacunas	15
Tabla 8 Registrar pacientes	16
Tabla 9 Lista de vacunas según la edad del paciente	17
Tabla 10 Visualizar reportes del parte diario	18
Tabla 11 Registrar cantidad de vacunas a enviar	18
Tabla 12 Resultados de las pruebas de caja negra del sistema	55
Tabla 13 Resultados de las pruebas de carga del sistema	61
Tabla 14 Resultados del test de carga con número mínimo de usuarios	61
Tabla 15 Resultados del test de carga con el doble de usuarios	63
Tabla 16 Resultados del test de stress del sistema	65

ÍNDICE DE FIGURAS

Figura 1 Proceso para Registro de Vacunas	6
Figura 2 Ingreso al sistema	20
Figura 3 Gestión de usuarios.....	21
Figura 4 Registrar vacuna	21
Figura 5 Inventario de vacunas	22
Figura 6 Registro de pacientes.....	22
Figura 7 Registrar vacunas aplicadas al paciente	23
Figura 8 Visualizar reportes.....	23
Figura 9 Registrar cantidad de vacunas a enviar	24
Figura 10 Secuencia Ingreso al Sistema	25
Figura 11 Secuencia gestión de usuarios	26
Figura 12 Secuencia registro de vacunas	27
Figura 13 Secuencia visualizar inventario de vacunas	28
Figura 14 Secuencia registrar paciente	29
Figura 15 Secuencia registrar vacuna aplicada al paciente.....	30
Figura 16 Secuencia visualizar reportes	31
Figura 17 Secuencia Banco de Vacunas	32
Figura 18 Diagrama conceptual módulo enfermería	33
Figura 19 Diagrama conceptual registro de paciente.....	34
Figura 20 Diagrama conceptual configuración biológico.....	35
Figura 21 Diagrama conceptual solicitud de vacuna	36

Figura 22 Diagrama conceptual banco vacuna	37
Figura 23 Modelo MCV utilizado en el Sistema	38
Figura 24 Diagrama de despliegue	39
Figura 25 Diagrama físico de la Base de Datos	40
Figura 26 Código para consultar la cantidad sobrante del mes anterior	41
Figura 27 Código para sumar la dosis del saldo anterior y la cantidad recibida del mes actual...	42
Figura 28 Código para registrar las vacunas ingresadas en el reporte Kardex	42
Figura 29 Código para elegir opciones y consultar datos sobre el saldo anterior y cantidad recibida.....	43
Figura 30 Código para clasificar el tipo de identificación	44
Figura 31 Código para generar el código de 17 dígitos	45
Figura 32 Código para insertar datos en captación temprana	46
Figura 33 Código para insertar la dosis en captación temprana	47
Figura 34 Código para seleccionar la dosis de la vacuna administrada.....	48
Figura 35 Código para desplegar los bloques de vacunas	49
Figura 36 Código para desplegar los bloques de vacunas según los parámetros de edad del paciente	50
Figura 37 Código general para consultar las dosis disponibles	51
Figura 38 Código para calcular la edad del paciente	51
Figura 39 Código para actualizar las cantidades disponibles si hubo desperdicios.....	52
Figura 40 Código para validar que vacunas tienen desperdicios y consultar las dosis.....	53
Figura 41 Código para consultar el consolidado mensual de la captación temprana	54

RESUMEN

El presente documento expone el desarrollo de un sistema informático que administra dos módulos, el de bodega donde consta los procesos de envío, recepción, solicitud mensual y un consolidado de dosis administradas, a su vez se encuentra el módulo para enfermería el cual registra los pacientes que acceden a una vacuna determinada según el esquema de edades, este sistema funcionará como herramienta para la correcta gestión de la información automatizando los formularios durante el movimiento de vacunas.

En el área de vacunación dentro del Centro de Salud del Centro Histórico realiza los procesos de recepción de vacunas, registro de dosis administradas además se procede a registrar pacientes para administrar vacunas según su edad el cual se realiza mediante un formulario en papel denominado parte diario y los informes se presentan en hojas de cálculo y de forma física, lo que ha provocado pérdida de información, duplicidad de datos y demora en los procesos que se manejan en esta área.

Este proyecto se llevó a cabo con la metodología Scrum ya que ha sido la que más se adaptó a las necesidades y características del proyecto porque permite comunicación constante con el cliente, trabajo en equipo y asignar tareas lo que facilitó el avance del sistema.

El resultado final fue un sistema que cumple con los requerimientos del Centro de Salud en el proceso de registro, administración y visualización de reportes, siendo capaz de procesar de manera adecuada hasta 396 usuarios concurrentes como resultado de las pruebas de carga y stress.

ABSTRACT

This document presents the development of a computer system that manages two modules, the warehouse where the processes of shipping, receiving, monthly request and a consolidated dose administered, in turn is the module for nursing which records patients accessing a particular vaccine according to the age scheme, this system will function as a tool for the proper management of information by automating the forms during the movement of vaccines.

The vaccination area at the Health Center of Centro Histórico carries out the processes of receiving vaccines, recording the doses administered and registering patients to administer vaccines according to their age, which is done using a paper form called a daily report and the reports are presented in spreadsheets and physically, which has caused loss of information, duplication of data and delays in the processes that are handled in this area.

This project was carried out with the Scrum methodology since it has been the one that best adapted to the needs and characteristics of the project because it allows constant communication with the client, teamwork and assigning tasks, which facilitated the progress of the system.

The final result was a system that meets the requirements of the Health Center in the process of registration, administration and visualization of reports, being able to adequately process up to 396 concurrent users as a result of load vs stress tests.

INTRODUCCIÓN

Antecedentes

“La administración de la información a llegado a ser un factor muy importante para las organizaciones que buscan automatizar procesos por medio de sistemas de información pues optimizan actividades cotidianas y son objeto de toma de decisiones” (FERRER, 2015)

El ministerio de Salud Pública del Ecuador lleva a cabo un proceso de vacunación el cual consiste en que el banco de vacunas envía suministros médicos a las diferentes unidades del distrito que le corresponde donde el personal de bodega recepta por medio de un registro llamado Kardex para tener constancia de las dosis de dicho mes para poder administrar a los pacientes en los centros de salud. El personal médico maneja un formulario donde se registra datos del paciente y la respectiva vacuna según la captación que depende de la edad del paciente. Para finalizar los encargados de bodega realizan una solicitud e informes al final de cada mes para continuar con el proceso de movimiento de vacunas.

El propósito de este proyecto es automatizar los formularios utilizados en las áreas de bodega y enfermería dentro del centro de salud los cuales generarán reportes con datos que podrán analizarse estadísticamente para la cobertura de vacunas a nivel de población en la ciudad de Quito.

El sistema constará de dos módulos correspondientes a las áreas ya mencionadas las cuales registrarán por cada mes el movimiento de vacunas en el establecimiento además se asignarán usuarios que podrán acceder a la aplicación para ejecutar funciones de registros en formularios y visualizaciones de reportes. Adicionalmente se incluirá informes estadísticos con gráficos acerca de la cobertura de vacunas en la población ecuatoriana.

Justificación

Las necesidades actuales de información, para la toma de decisiones, el conocimiento de la situación de una institución de salud, el manejo de sus datos epidemiológicos, y su organización administrativa, hacen que quienes se encuentran comprometidos con estos estamentos deban conocer, como mínimo, las nociones de lo que es un sistema de información, y tener una idea básica de lo que la informática puede hacer en estos ámbitos (Salamea, 2003).

En este sentido el sistema propuesto es un factor de suma importancia para la optimización del registro de información de vacunas y disminución del tiempo para la consulta de información en los módulos de bodega y enfermería. Esta información será oportuna, verídica y accesible para los usuarios cuando la necesiten. Este sistema también ofrece diversos reportes de datos con la frecuencia diaria, semanal y/o mensual.

Con el desarrollo de este sistema finalmente se solucionarán los problemas como la inconsistencia de datos que se generan con los procesos manuales de registros y el tiempo que se gasta al corregir estos errores en los informes mensuales. Con el fin de que la información sea más fiable y pueda ser analizada para la toma de decisiones oportuna.

El grupo que será beneficiado con este proyecto será el personal encargado de las respectivas áreas de bodega de suministros médicos y las enfermeras del centro de salud quienes tienen la labor de ingresar datos del paciente, administrar la respectiva vacuna y realizar los informes mensuales.

Objetivos

Objetivo General

Desarrollar un sistema informático para eliminar la inconsistencia de datos en los módulos de bodega y enfermería durante el movimiento de vacunas utilizados en el Centro de salud tipo C del Centro Histórico.

Objetivos Específicos

Analizar los requerimientos funcionales y no funcionales para el registro de formularios de vacunas en los módulos de bodega y enfermería.

Aplicar la metodología Scrum para el proceso y evidenciar la operatividad del sistema de información empleando además artefactos UML.

Desarrollar el sistema informático en base al diseño proporcionado a través de la metodología.

Realizar pruebas de operatividad que aseguren la adecuada funcionalidad del sistema.

Marco Metodológico

Para el desarrollo de software existen numerosas alternativas de metodologías que se llevan a cabo con diferentes dimensiones del proceso de desarrollo.

Estas metodologías se dividen en dos tipos:

- **Metodología Tradicional:** Se prioriza en base a la visión propia de la dirección del proyecto y no siempre en base a la visión del negocio.
- **Metodología Ágil:** Trabaja con entregas incrementales e integrales continuas estableciendo un proceso sobre básicos, en esta metodología se puede reencaminar el

proyecto rápidamente cuando las cosas no estén funcionando como teóricamente estaba previsto en el inicio del proyecto.

La tabla 1 compara las distintas metodologías ágiles basándose en parámetros como: colaboración del equipo, cambios continuos del sistema y resultados finales. También se ha incluido la metodología tradicional Kanban para comparar sus características.

Tabla 1

Comparación de Metodologías SCRUM, XP, KANBAN

CARACTERÍSTICAS	METODOLOGÍAS		
	SCRUM	XP	KANBAN
Metodología flexible	x	x	
Trabajo en equipo, aprendizaje continuo y buen clima de trabajo.	x	x	
Existe una interacción constante con el cliente y el equipo de desarrollo	x	x	
Adaptación a los cambios y sus respectivas configuraciones	x		
El equipo de desarrollo es capaz de autoorganizarse	x		
El equipo trabaja con roles donde se encarga de distintas tareas.	x		x
Entrega continua del proyecto.	x		x
Hay una rápida respuesta en base a los cambios	x	x	x

Elaborado por Wendy German y Dorival Pichamba

En base a la Tabla 1, la metodología SCRUM es la metodología que se aplicará para llevar a cabo este proyecto.

CAPÍTULO 1 ESTADO DEL ARTE

1.1 Marco Institucional

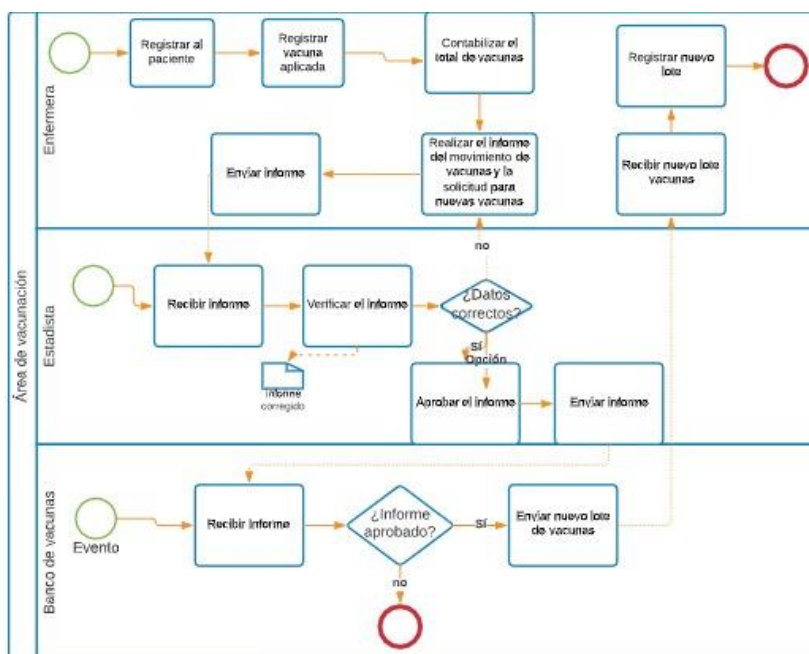
El presente proyecto se realizará en el centro de salud tipo c “Centro Histórico” correspondiente al primer nivel de atención de salud que pertenece al Ministerio de Salud Pública dentro del distrito 17D04 zona 9.

La misión del Ministerio de Salud Pública es

Ejercer la rectoría, regulación, planificación, coordinación, control y gestión de la Salud Pública ecuatoriana a través de la gobernanza y vigilancia y control sanitario y garantizar el derecho a la Salud a través de la provisión de servicios de atención individual, prevención de enfermedades, promoción de la salud e igualdad, la gobernanza de salud, investigación y desarrollo de la ciencia y tecnología; articulación de los actores del sistema, con el fin de garantizar el derecho a la Salud. (Ministerio de Salud Pública, 2021)

El movimiento de vacunas empieza desde un Banco Zonal donde se encuentra a cargo el Lic. Juan Carlos Cruz quien es el que recepta las solicitudes y coordina el envío de suministros a los respectivos distritos asignados. En el centro de salud tipo C donde se realiza el proyecto la Lic. Karol Paredes se encarga del módulo de bodega y enfermería donde realizan la recepción y administración de vacunas llenando formularios denominados “parte diarios” de 3 captaciones temprana, tardío y dT Pediátrico. Al finalizar se realiza una solicitud para el siguiente mes donde consta datos de vacunas totales y sobrantes que van dirigidos al Banco Zonal para enviar nuevos lotes al distrito además dichos informes son enviados al estadista Giovanni Andino quien se encarga de revisar el proceso de vacunación.

En la figura 1 se describe el proceso para el registro de vacunas dentro del centro de salud.

Figura 1*Proceso para Registro de Vacunas*

Elaborado por Wendy German y Dorival Pichamba

Dentro del proceso que se describe en la figura 1 el desarrollo del sistema esta centrado en dar solución a la automatizacion del formulario de registro de pacientes y la generación de informes sobre el moviento de las vacunas.

1.2 Marco Teórico

1.2.1 Sistema Informático

“Conjunto formal de procesos que, operando sobre una colección de datos estructurada de acuerdo a las necesidades de la empresa, recopila, elabora y distribuyen selectivamente la información necesaria para la operación de dicha empresa y para las actividades de dirección y control correspondientes, apoyando, al menos en parte, los procesos de toma de decisiones

necesarios para desempeñar funciones de negocio de la empresa de acuerdo con su estrategia” (Andreu, Ricart, & Valor, 1991).

1.2.2 Metodología Scrum

Según (Laínez, 2014) Scrum es un proceso para desarrollar software incrementalmente en entornos complejos donde los requisitos no están claros o cambian con mucha frecuencia”. Se ha decidido utilizar esta metodología porque es abierta y flexible es decir que puede adaptarse a las necesidades del cliente además que como equipo de desarrollo se puede organizar y asignar tareas para poder dar como resultado un sistema informático eficiente, seguro y de calidad.

1.2.2.1 Planificación del Sprint

Es donde se realiza la primera reunión del equipo se definirán aspectos como la funcionalidad, objetivos, riesgos del sprint, plazos de entrega, entre otros. Posteriormente se realiza una junta entre el equipo y el jefe del proyecto para explicar cómo se desarrollará cada punto del intervalo. (Business School, 2018)

1.2.2.2 Etapa de Desarrollo

“Los encargados deben garantizar que no se generen cambios de último momento que puedan afectar los objetivos del mismo. Además, se asegura el cumplimiento de los plazos establecidos para su término” (Business School, 2018).

1.2.2.3 Revisión del Sprint

“Al final del desarrollo del intervalo, es posible analizar y evaluar los resultados. Si es necesario, todo el equipo colaborará para saber qué aspectos necesitan ser cambiados. En esta fase se fomenta la colaboración y retroalimentación entre todos” (Business School, 2018).

1.2.2.4 Retroalimentación

“Los resultados pueden ser entregados para recibir un feedback no solo por parte de los profesionales dentro del proyecto, sino también de las personas que utilizarán directamente lo que se desea lograr; es decir, los clientes potenciales” (Business School, 2018).

1.2.3 Herramientas de Desarrollo

1.2.3.1 Servidor HTTP Apache

“Apache es un servidor HTTP de código abierto multiplataforma. Su objetivo es proveer un servidor seguro, eficiente y extensible que proporcione servicios HTTP en sincronía con los estándares HTTP actuales” (Grupo D, 2020).

Características:

- Es multiplataforma está disponible en Windows, Mac, Unix y Linux.
- Es gratuito y Open Source.

(Grupo D, 2020)

1.2.3.2 Hypertext Preprocessor (PHP)

“Es un lenguaje de scripting de código abierto de uso general popular que es especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML” (APR, 2021). PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos” (The PHP Group, 2021).

Con PHP, se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas.

Características:

- Es de código abierto, no hay restricciones de uso vinculadas a los derechos. El usuario puede usar PHP para programar en cualquier proyecto y comercializarlo sin problemas.
 - Ayuda a disminuir el tiempo de carga de las páginas, permitiendo que el servidor trabaje con más suavidad para cargar plugins y aplicaciones en los sitios web.
 - Desarrollar con agilidad sitios con un gran rendimiento, incluso si están llenos de recursos, y con la garantía de la sostenibilidad del desempeño a largo plazo.
 - Admite una gran cantidad de datos.
- (The PHP Group, 2021)

1.2.3.3 JavaScript

JavaScript es un lenguaje de programación del lado del cliente que permite implementar dinamismo y funcionalidad a las páginas web.

Cuando JavaScript se ejecuta en el navegador, no necesita de un compilador. El navegador lee directamente el código, sin necesidad de terceros. Por tanto, se le reconoce como uno de los tres lenguajes nativos de la web junto a HTML (contenido y su estructura) y a CSS (diseño del contenido y su estructura) (MDN, 2021)

Características:

- Multiplataforma, se puede utilizar en Windows, Linux o Mac o en el navegador de preferencia.
 - Es Imperativo y estructurado, mediante un conjunto de instrucciones indica al computador qué tarea debe realizar.
 - Prototipado, debido a que usa prototipos en vez de clases para el uso de herencia.
 - Es dinámico, responde a eventos en tiempo real.
- (MDN, 2021)

1.2.3.4 Scripcase

ScriptCase es una poderosa herramienta RAD (Rapid Application Development) que genera aplicaciones PHP basadas en bases de datos.

Con esta herramienta se puede crear Sistemas web completos, y también se puede generar Grids, Filtros, Gráficas, informes en PDF, Formularios, menús de navegación, exportación de datos a DOC, XLS, CSV, y XML, módulos de seguridad con login y niveles de usuarios y grupos.

El lenguaje esencial que maneja es PHP del lado del servidor y hace uso de otras tecnologías como AJAX, JavaScript, HTML y CSS para el lado del cliente brindando dinamismo y mejorando el aspecto de la página.

Para el desarrollo de proyectos Scriptcase, implementa una arquitectura de desarrollo basada en “MVC (Modelo-Vista-Controlador)” mismo que facilita el manejo de contenido, haciendo sistemas más seguros y con tiempos de respuesta cortos (Scriptcase blog, 2019).

Características

- Es conformado por código fuente php, JavaScript, HTML y Ajax.
- Funciona en todo servidor web compatible con php.
- Se puede generar aplicaciones web directamente a partir de la base de datos, permitiendo ahorrar tiempo, reducir costos y aumentar su productividad.
- Permite crear reportes con filtros por medio de búsquedas iniciales en el despliegue de este.

(Scriptcase blog, 2019)

CAPÍTULO 2 ANÁLISIS Y DISEÑO

2.1 Análisis de Requerimientos

2.1.1 Requerimientos Funcionales

Tabla 2

Requerimientos Funcionales

Código	Requerimiento
RF1	El sistema iniciara con una pantalla de login. Ninguno
RF2	El sistema permitirá crear, editar y eliminar usuarios, sus permisos y perfiles.
RF3	El sistema debe registrar las vacunas de los establecimientos del mismo distrito.
RF4	El sistema generará un inventario (Kardex) de todas las vacunas.
RF5	El Sistema debe registrar los datos personales del paciente.
RF6	El sistema debe desplegar las vacunas según la edad del paciente, basándose en el esquema de vacunación.
RF7	El sistema permitirá visualizar reportes diarios del registro de pacientes.
RF8	El sistema permitirá ingresar cantidades a enviar de dosis a los establecimientos asignados además de realizar un registro mensual con las vacunas y establecimientos.

Elaborado por Wendy German y Dorival Pichamba

2.1.2 Requerimientos no Funcionales

Tabla 3

Requerimientos no Funcionales

Código	Requerimiento
RNF0	El sistema debe utilizarse con los navegadores web Chrome, Firefox e Internet Explorer.
RNF1	Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden
RNF2	El sistema permitirá la consulta a la base de datos en tiempo real.
RNF3	El sistema debe presentar una interfaz sencilla e intuitiva para el usuario.

Elaborado por Wendy German y Dorival Pichamba

2.2 Historias de Usuario

Las historias de usuario son uno de los componentes centrales de un programa ágil. Ayudan a proporcionar un marco centrado en el usuario para el trabajo diario, lo que impulsa la colaboración y la creatividad y mejora el producto en general (Rehkopf, 2018).

Acceso al Sistema

Tabla 4

Inicio de Sesión

Actores	Administrador
Requisito Funcional	RF1
Flujo Normal	<p>El actor accede al login del sistema e ingresa sus credenciales: usuario y contraseña.</p> <p>El sistema comprueba los datos ingresados.</p> <p>El sistema inicia la sesión del usuario registrado.</p>

Flujo Alternativo	<p>2. a. El sistema detecta un error en el campo usuario y emite el error de “Usuario incorrecto”</p> <p>2. b. El sistema detecta un error en el campo clave y emite el error de “Contraseña incorrecta”.</p>
--------------------------	---

Elaborado por Wendy German y Dorival Pichamba

Tabla 5 *Crear Usuarios y Asignar Roles*

Actores	Administrador
Requisito Funcional	RF2
Flujo Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la interfaz de usuarios. 2. El sistema presenta en pantalla una lista de los usuarios ingresados. 3. El actor selecciona: <ol style="list-style-type: none"> a. añadir <ol style="list-style-type: none"> i.El sistema presenta una pantalla con los campos requeridos para el ingreso de un usuario al sistema. ii.El actor ingresa la información solicitada, selecciona el perfil de usuario y selecciona la opción guardar. iii.El sistema presentará en pantalla un mensaje de confirmación del usuario guardado. b. Editar <ol style="list-style-type: none"> i.El sistema muestra en pantalla los datos del usuario seleccionado. ii.El actor edita los campos del usuario que se requiera y selecciona la opción guardar. iii.El sistema en respuesta presenta un mensaje en pantalla con la confirmación de que los datos se han guardado. c. Eliminar <ol style="list-style-type: none"> i.El actor selecciona al usuario que se va a eliminar. ii.El actor confirma la operación. iii.El sistema refresca la lista de usuarios. 4. El sistema refresca la lista de usuarios.
Flujo Alternativo	<p>3a. En caso de estar vacío el campo de usuario, contraseña, confirmar contraseña y e-mail emitirá el error de “Campo obligatorio”.</p>

Elaborado por Wendy German y Dorival Pichamba

Tabla 6

Registrar vacunas de los establecimientos

Actores	Encargado de recepción de vacunas
Requisito Funcional	RF3
Flujo Normal	<ol style="list-style-type: none"> 1. El flujo empieza cuando el actor accede al sistema. 2. El sistema mostrará en pantalla un menú. 3. En el menú el actor selecciona bodega. 4. El sistema en respuesta despliega en pantalla las opciones dentro de bodega. 5. El actor selecciona "Recepción de vacunas". 6. El actor ingresa la fecha de entrega 7. El sistema presenta en pantalla la lista de los establecimientos del distrito. 8. El actor selecciona el establecimiento en el que se va a registrar la vacuna. 9. El sistema presenta en pantalla la lista de vacunas que se encuentran registradas en ese establecimiento. 10. El actor selecciona agregar biológico. 11. El sistema muestra la interfaz con los campos requeridos para el registro de vacunas. 12. El actor llena los campos requeridos para el registro. 13. El actor selecciona guardar. 14. En respuesta el sistema muestra en pantalla un mensaje de confirmación indicando que la vacuna se ha registrado correctamente. 15. El sistema actualiza la lista de vacunas.
Flujo Alternativo	<ol style="list-style-type: none"> 1a. El sistema detecta las credenciales incorrectas y emite el error "Usuario o contraseña incorrectos" 10a. El sistema detecta que no ha sido ingresado el campo obligatorio y emite el error "Campo obligatorio" 11a. El sistema detecta el campo vacío y emite el error "Campo obligatorio" 12a. El sistema detecta el campo vacío y emite el error de "Campo obligatorio"

Elaborado por Wendy German y Dorival Pichamba

Tabla 7

Visualizar inventario de las vacunas

Actores	Enfermera
Requisito Funcional	RF4
	<ol style="list-style-type: none"> 1. El actor accede al sistema. 2. El sistema despliega en pantalla el menú general. 3. El actor selecciona la opción "Reportes" 4. El sistema muestra un submenú con las categorías de los reportes. 5. El actor selecciona la categoría "Partes Diarios" y tiene acceso a la Captación Temprana, Tardía y Dt. Pediátrico 6. En respuesta el sistema presenta en pantalla la visualización del reporte con filtros de búsqueda. 7. El actor selecciona la categoría "Informe de vacunas" y tiene acceso a Kardex Mensual y Kardex Diario.
Flujo Normal	<ol style="list-style-type: none"> 8. En respuesta el sistema presenta en pantalla la visualización del reporte con filtros de búsqueda. 9. El actor selecciona la categoría "Reporte Solicitud de vacunas" 10. En respuesta el sistema presenta en pantalla la visualización del reporte con filtros de búsqueda. 11. El actor selecciona la categoría "Reportes Generales" y tiene acceso a los reportes de Captación temprana general, captación tardía general y captación dt general. 12. En respuesta el sistema presenta en pantalla la visualización del reporte con filtros de búsqueda.
Flujo Alternativo	<ol style="list-style-type: none"> 5a. En caso de no existir registro emite el error "No hay registros para mostrar"

- 7a. En caso de no existir registro emite el error “No hay registros para mostrar”
- 9a. En caso de no existir registros la categoría emitirá el siguiente mensaje “No hay registros para mostrar”
- 11a. En caso de no existir registros la categoría emitirá el siguiente mensaje “No hay registros para mostrar”

Elaborado por Wendy German y Dorival Pichamba

Tabla 8

Registrar pacientes

Actores	Enfermera
Requisito Funcional	RF5
Flujo Normal	<ol style="list-style-type: none"> 1. El flujo empieza cuando el actor ingresa al sistema 2. El sistema mostrará los módulos 3. El actor seleccionará “Enfermería” en el menú 4. El sistema mostrará las opciones dentro del módulo 5. El actor elige la opción “Administración de vacunas” 6. En respuesta el sistema muestra la interfaz de registro de pacientes. 7. El actor registra los datos del paciente. 8. El sistema comprueba los datos y los guarda en la base de datos.
Flujo Alternativo	<p>4a. El sistema detecta que el campo de nombre, apellido, segundo nombre, segundo apellido, fecha de nacimiento, lugar de nacimiento se encuentra vacío y emite el error de “Campo vacío”</p> <p>4b. El sistema detecta que no se ha seleccionado nada en el campo “Documento de identidad” y emite el error de “Campo vacío”</p>

4c. El sistema detecta que el campo de cédula/pasaporte, lugar de residencia, provincia, cantón, parroquia, barrio se encuentra vacío y emite el error de “Campo vacío”

Elaborado por Wendy German y Dorival Pichamba

Tabla 9

Lista de vacunas según la edad del paciente

Actores	Enfermera
Requisito Funcional	RF6
Flujo Normal	<ol style="list-style-type: none"> 1. El flujo empieza cuando el paciente ya se encuentra registrado. 2. El sistema clasificará a los pacientes según su edad. 3. El sistema clasificará si el paciente pertenece a un grupo de riesgo 4. El sistema clasificará la captación de las vacunas según la edad. 5. El sistema mostrará las vacunas que correspondan a la clasificación. 6. El sistema mostrará las dosis correspondientes a cada vacuna.
Flujo Alternativo	<p>2a. El sistema detecta que no se ha ingresado nada en fecha de nacimiento por lo que emite el error “Campo obligatorio”</p> <p>4a. El sistema detecta que no se ha seleccionado nada en el campo de captación por lo que no mostrará ninguna vacuna.</p> <p>5^a. El sistema detecta que no ha seleccionado nada en el campo vacuna por lo que no mostrará ninguna dosis</p> <p>5b. El sistema verifica si la vacuna se encuentra disponible en caso no agotarse mostrará el error “Vacuna agotada”</p>

Elaborado por Wendy German y Dorival Pichamba

Tabla 10*Visualizar reportes del parte diario*

Actores	Enfermera
Requisito Funcional	RF7
Flujo Normal	<ol style="list-style-type: none"> 1. El actor accede a la interfaz de reportes. 2. El sistema muestra un submenú de las categorías de reportes que existen en el sistema. 3. El actor selecciona la categoría Reporte. 4. El sistema presenta en pantalla la lista de reportes. 5. El actor selecciona uno de ellos. 6. El reporte se visualiza en la pantalla con filtros de búsqueda. 7. El actor ingresa información requerida por el reporte. 8. Finalmente, el sistema muestra el resultado de la búsqueda.
Flujo Alternativo	<p>5a. Si el registro no existe emite el error de “No hay registros para mostrar.</p> <p>6ª Si le registro se realiza un filtro y no existe emite el siguiente mensaje “No hay registros para mostrar”</p>

Elaborado por Wendy German y Dorival Pichamba

Tabla 11*Registrar cantidad de vacunas a enviar*

Actores	Bodega
Requisito Funcional	RF8
Flujo Normal	<ol style="list-style-type: none"> 1. El actor ingresa al menú del sistema y elige la opción “Bodega” 2. El actor ingresa a la opción Envío de Vacunas 3. El actor ingresa la fecha de envío de vacunas 4. El actor selecciona el establecimiento al que se va a enviar 5. El actor selecciona que vacunas se va a enviar a dicho establecimiento

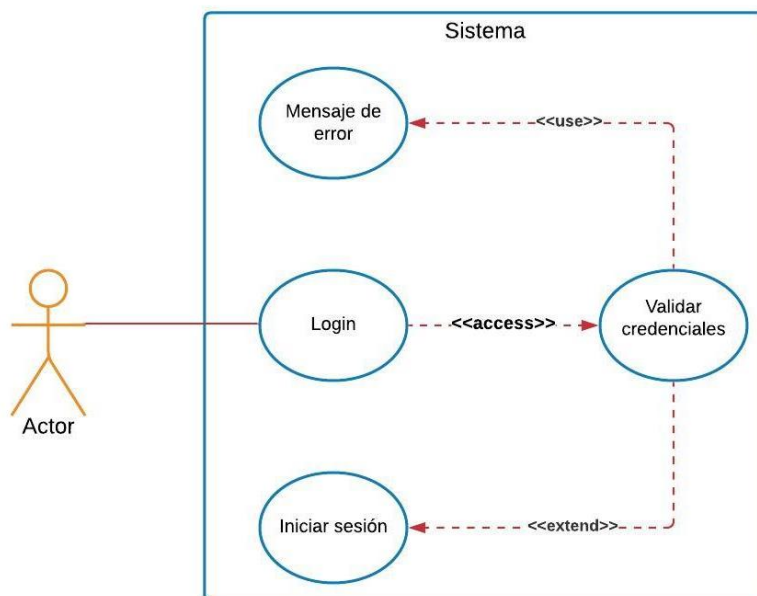
	6. El actor ingresa una cantidad mayor o igual a 0 para enviar las dosis al establecimiento asignado.
	7. El actor ingresa datos acerca del Banco de vacunas como provincias, cantón, parroquia, zona y distrito.
	8. El actor puede acceder a reportes y a la categoría "Informe del Banco de Vacunas" donde visualiza la fecha de envío y los establecimientos con sus vacunas.
Flujo Alternativo	3a. En caso de detectar un campo vacío el sistema emitirá el error "Campos obligatorios"
	6a, 7a En caso de detectar campos vacíos el sistema emitirá el error "Campos obligatorios"
	8a En caso de no existir registro el sistema emite el mensaje "No hay registros para mostrar"

Elaborado por Wendy German y Dorival Pichamba

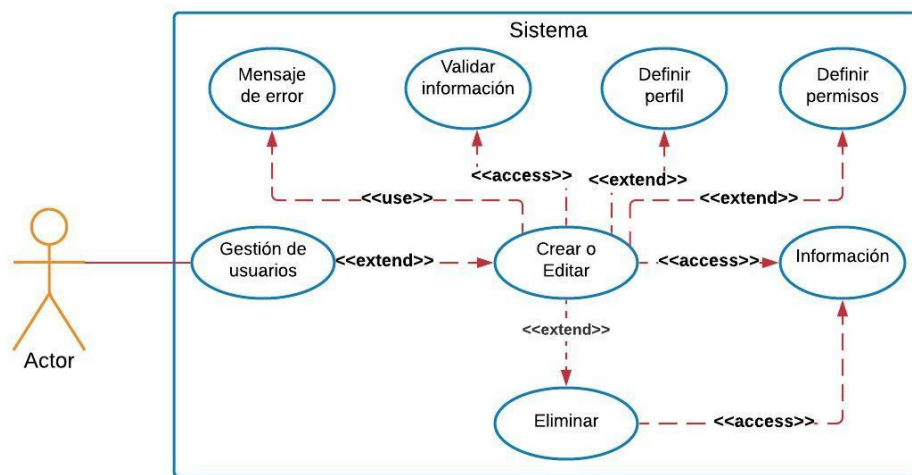
2.3 Diagramas de Casos de Uso

Los diagramas de casos de uso describen una acción o actividad y especifican la comunicación del sistema mediante la interacción de los usuarios.

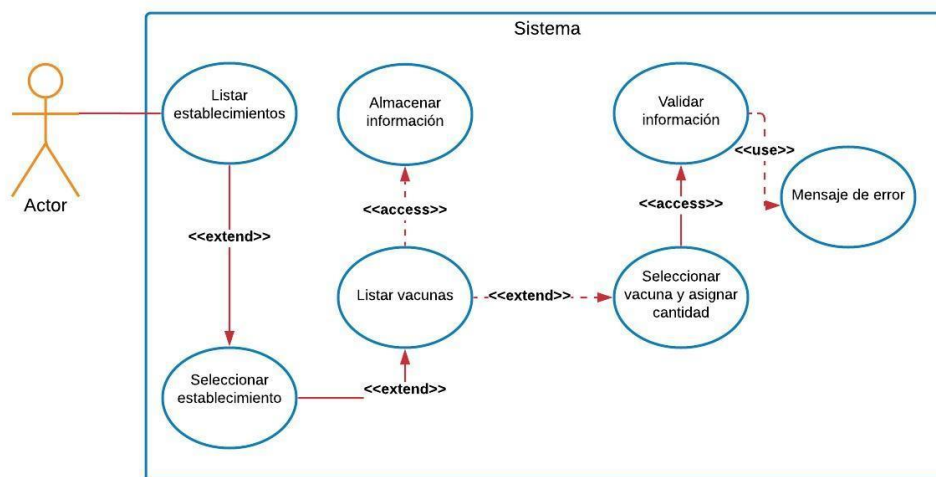
Ingreso al Sistema

Figura 2*Ingreso al sistema*

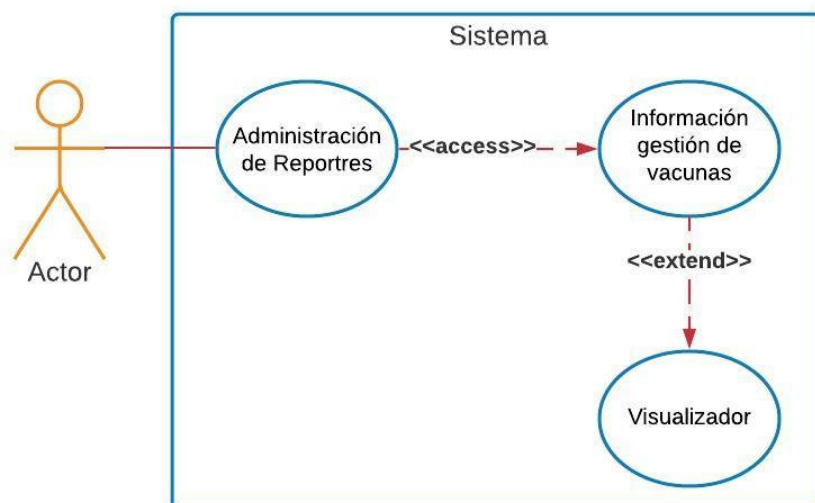
Elaborado por Wendy German y Dorival Pichamba

Figura 3*Gestión de usuarios*

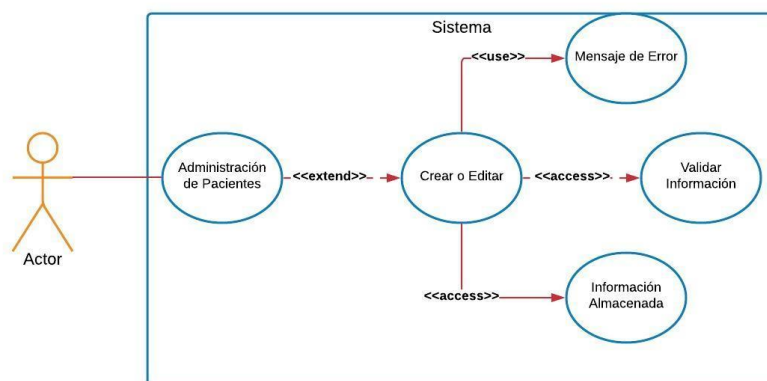
Elaborado por Wendy German y Dorival Pichamba

Módulo Bodega**Figura 4***Registrar vacuna*

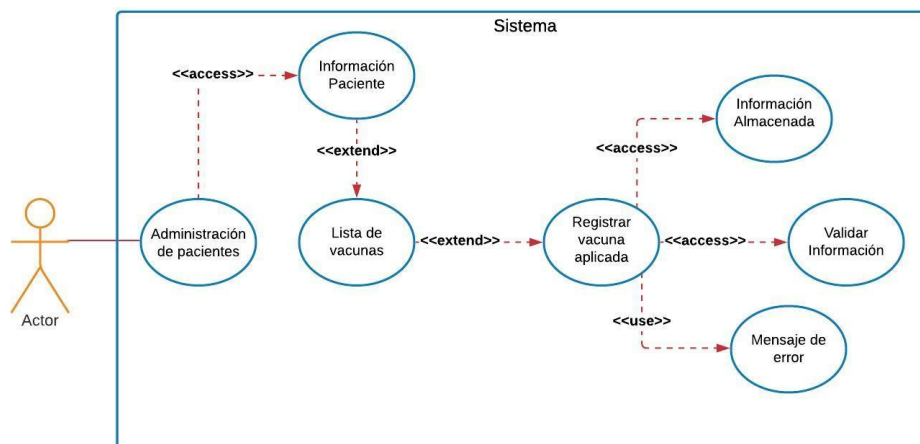
Elaborado por: Wendy German y Dorival Pichamba

Figura 5*Inventario de vacunas*

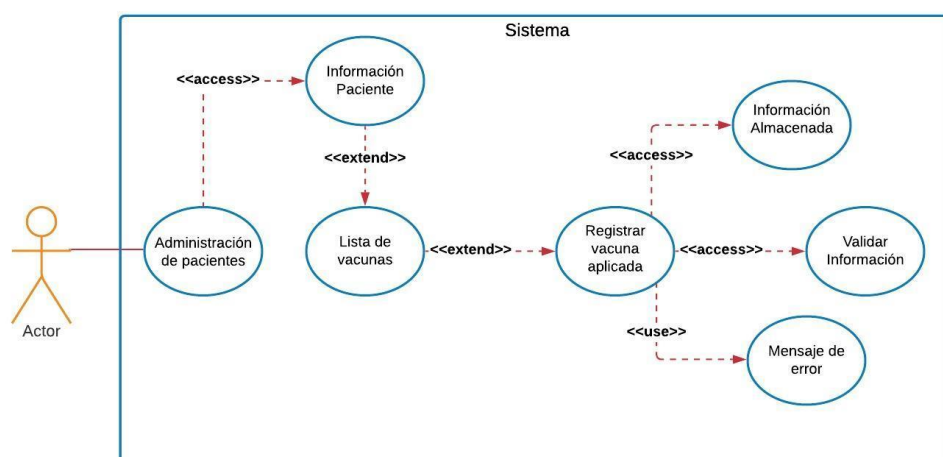
Elaborado por: Wendy German y Dorival Pichamba

Módulo Enfermería**Figura 6***Registro de pacientes*

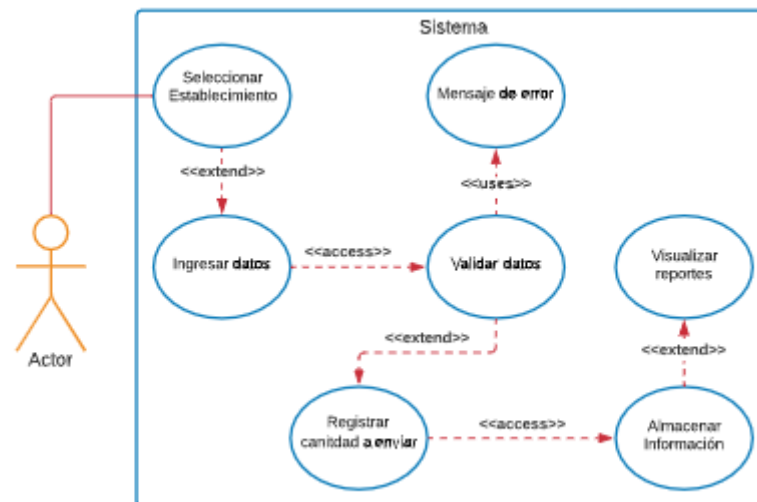
Elaborado por Wendy German y Dorival Pichamba

Figura 7*Registrar vacunas aplicadas al paciente*

Elaborado por: Wendy German y Dorival Pichamba

Figura 8*Visualizar reportes*

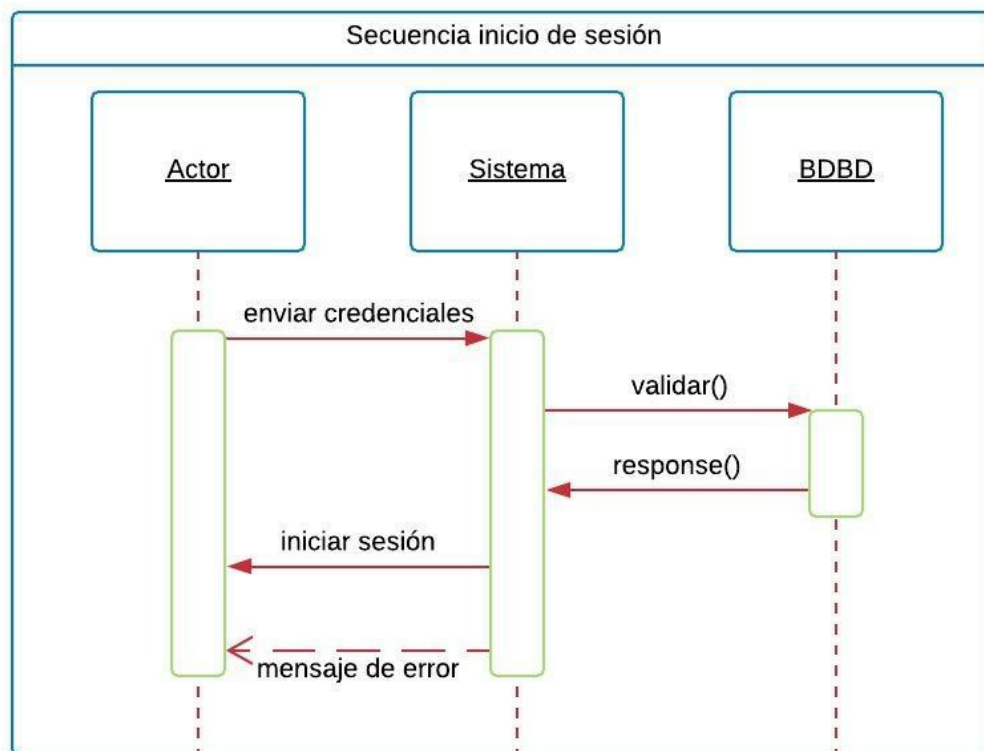
Elaborado por: Wendy German y Dorival Pichamba

Figura 9*Registrar cantidad de vacunas a enviar*

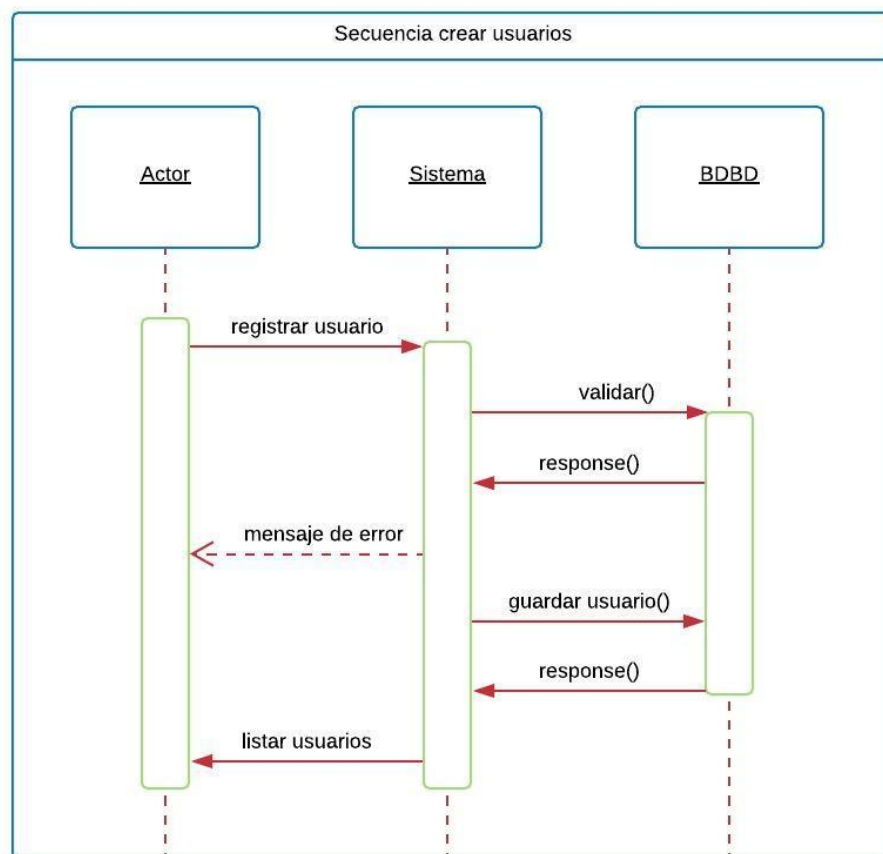
Elaborado por Wendy German y Dorival Pichamba

2.4 Diagramas de Secuencia

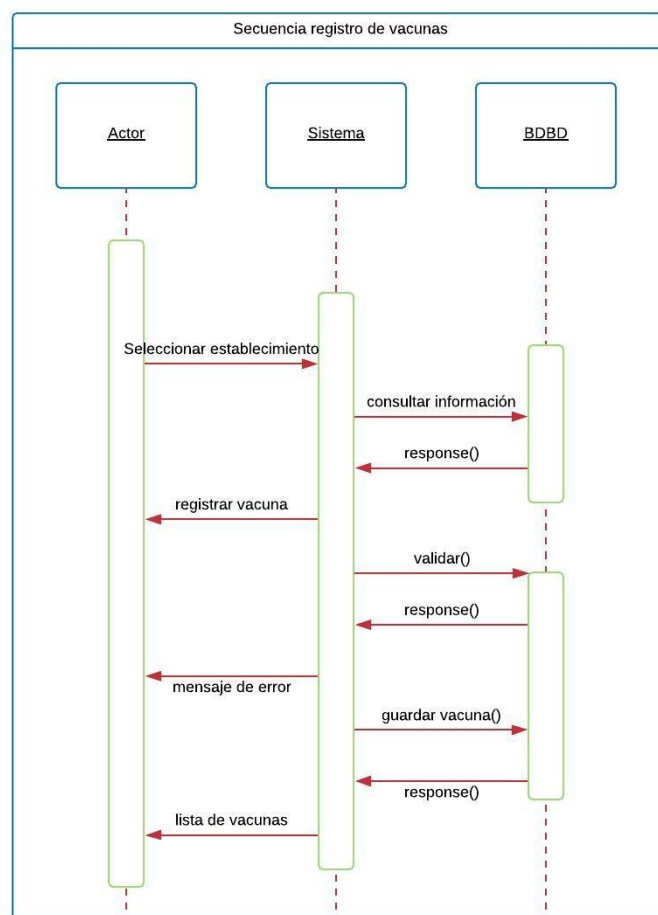
Los diagramas de secuencia muestran como interactúan los elementos del sistema desde el actor y sus procesos de manera ordenada.

Figura 10*Secuencia Ingreso al Sistema*

Elaborado por: Wendy German y Dorival Pichamba

Figura 11*Secuencia gestión de usuarios*

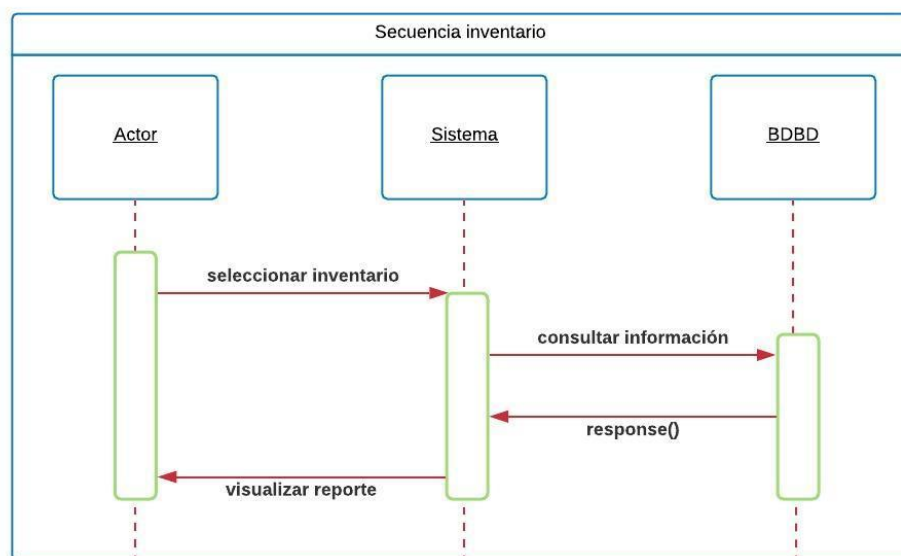
Elaborado por Wendy German y Dorival Pichamba

Figura 12*Secuencia registro de vacunas*

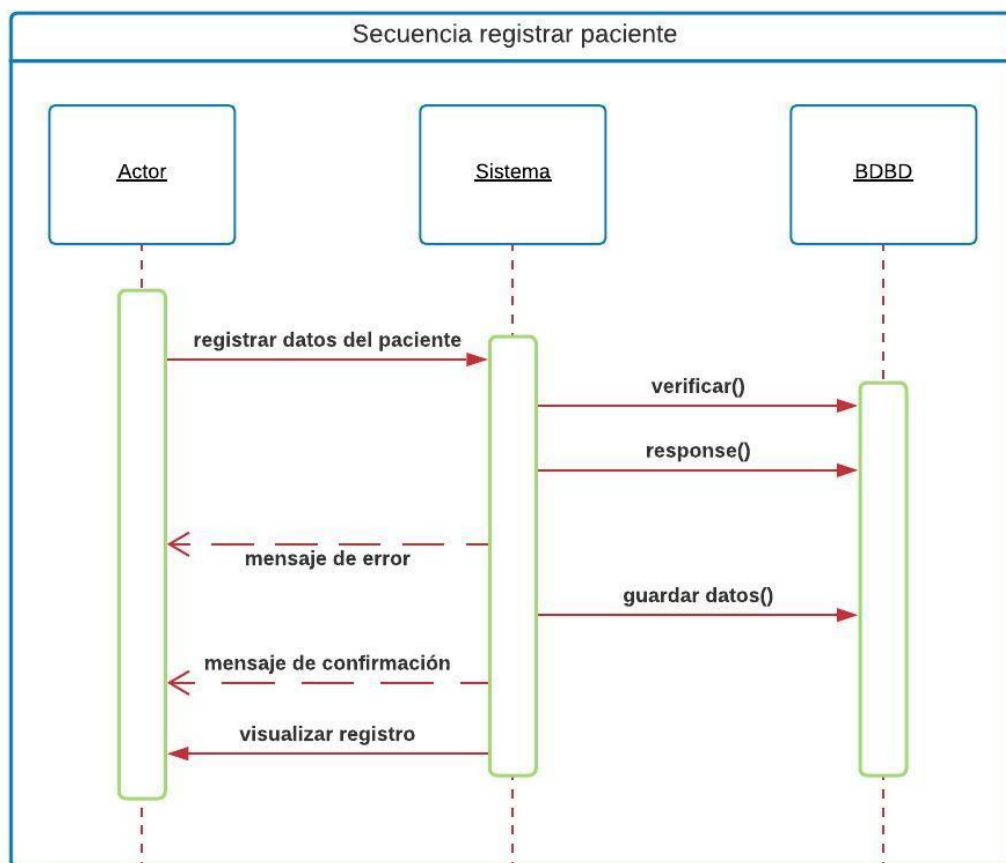
Elaborado por: Wendy German y Dorival Pichamba

Figura 13

Secuencia visualizar inventario de vacunas



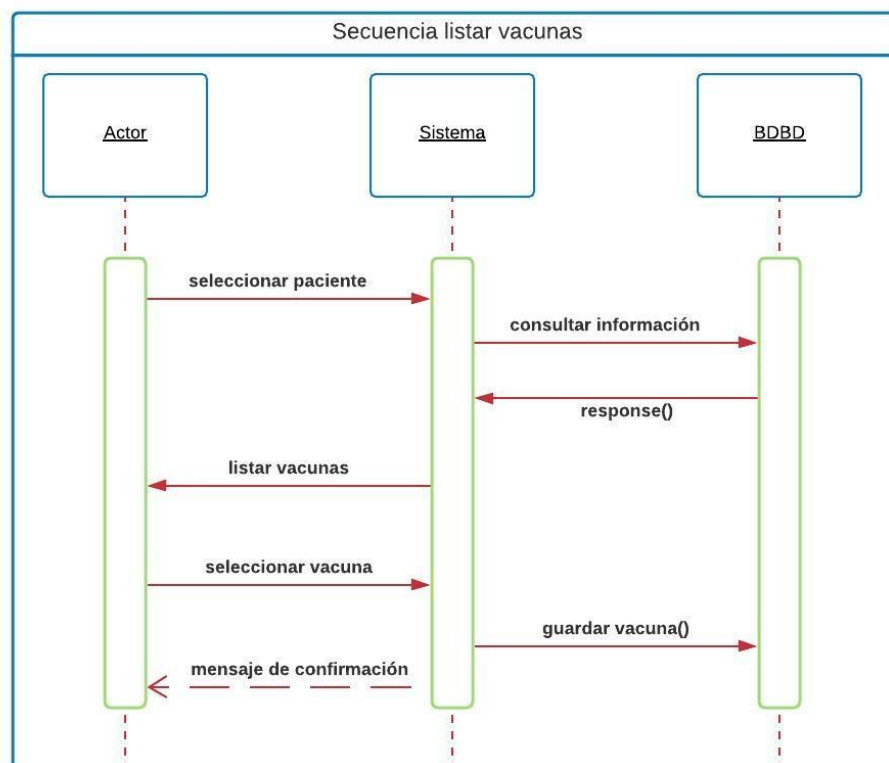
Elaborado por: Wendy German y Dorival Pichamba

Figura 14*Secuencia registrar paciente*

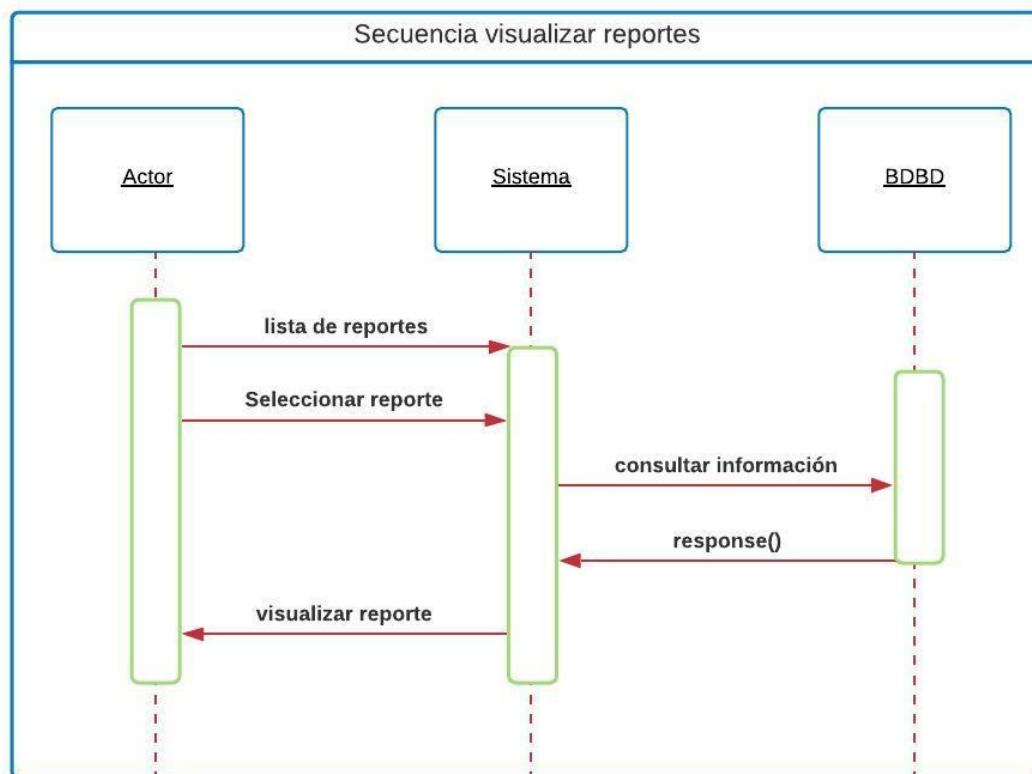
Elaborado por: Wendy German y Dorival Pichamba

Figura 15

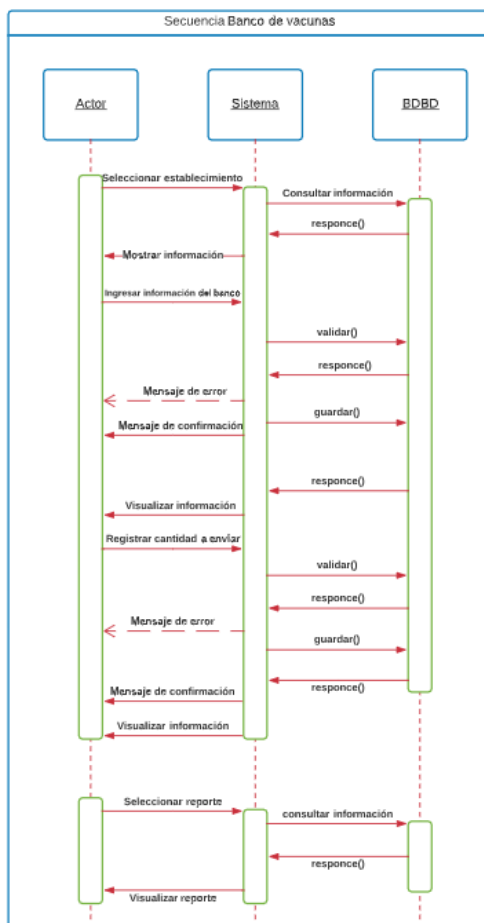
Secuencia registrar vacuna aplicada al paciente



Elaborado por: Wendy German y Dorival Pichamba

Figura 16*Secuencia visualizar reportes*

Elaborado por: Wendy German y Dorival Pichamba

Figura 17*Secuencia Banco de Vacunas*

Elaborado por Wendy German

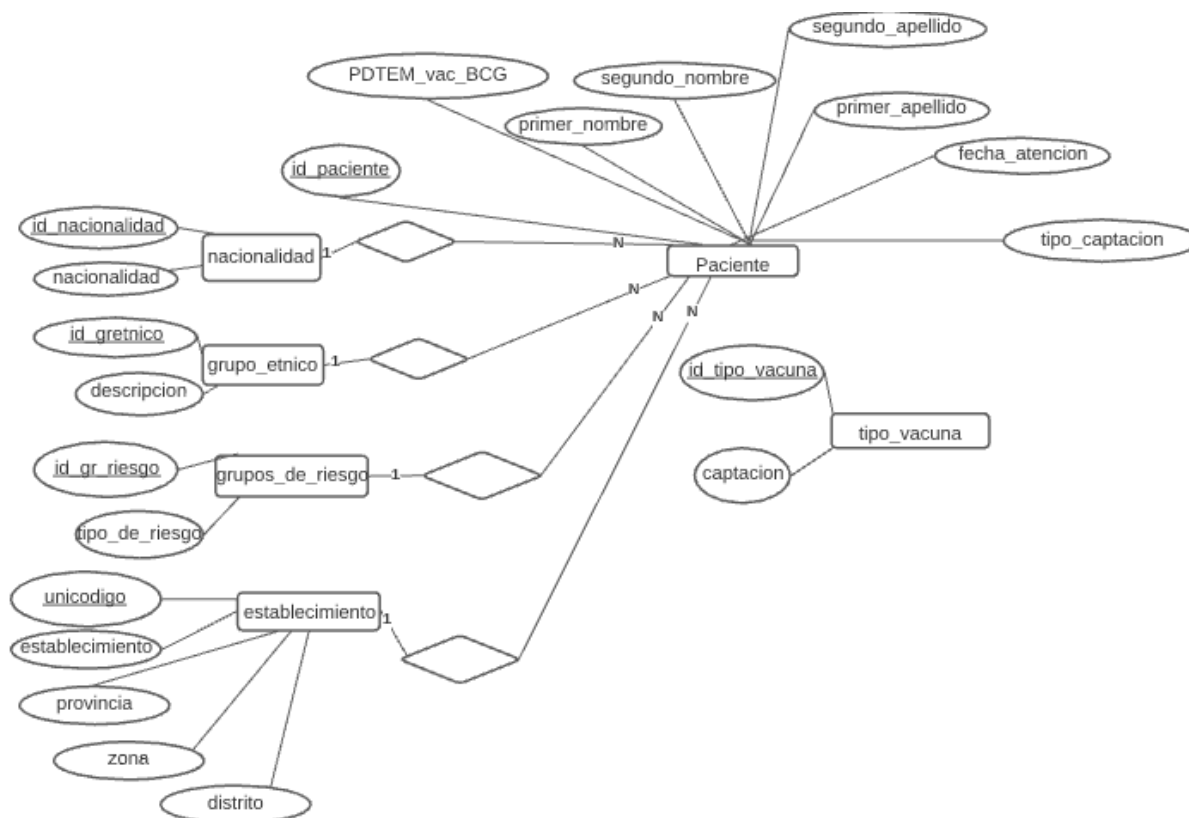
2.5 Diagrama Conceptual de la Base de Datos**Módulo Enfermería**

Figura 18*Diagrama conceptual módulo enfermería*

Elaborado por Wendy German y Dorival Pichamba

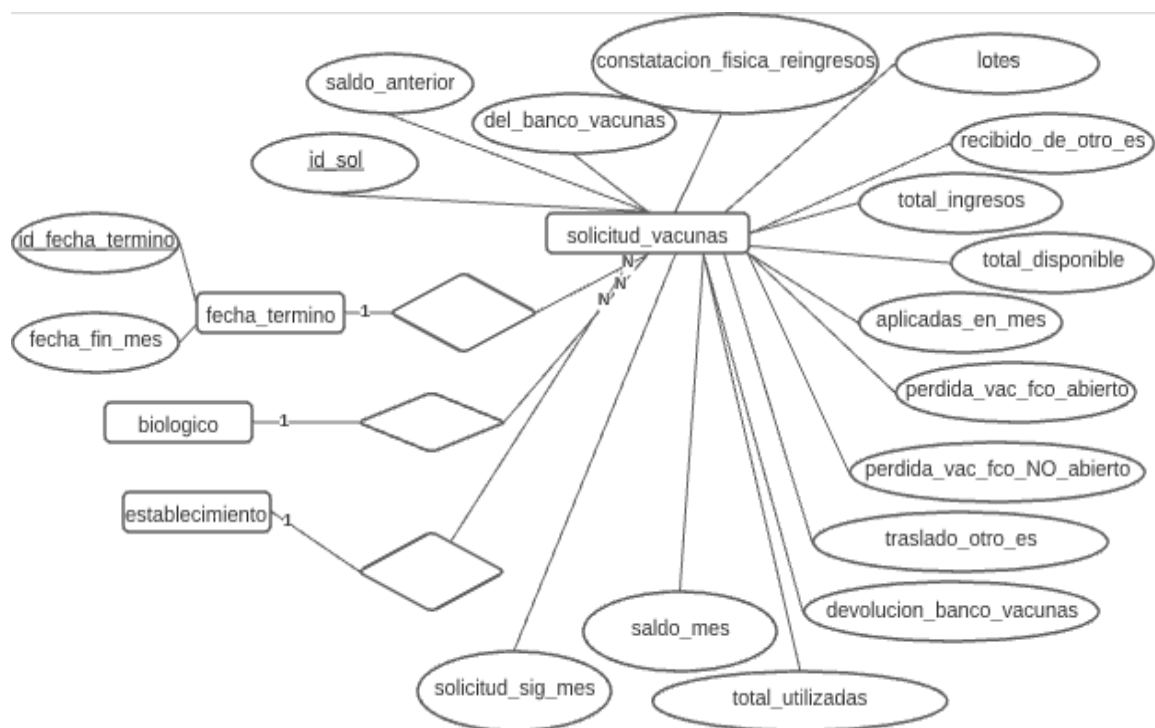
Figura 19

Diagrama conceptual registro de paciente.

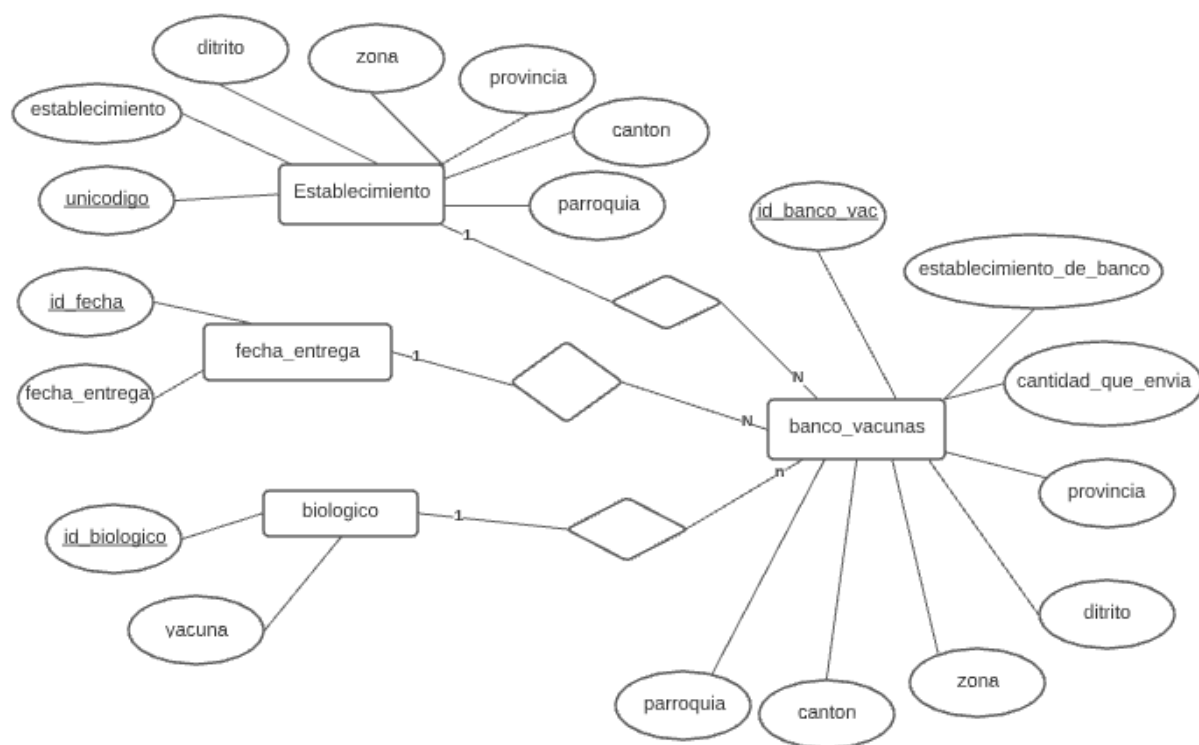


Elaborado por Wendy German y Dorival Pichamba

Módulo Bodega

Figura 21*Diagrama conceptual solicitud de vacuna*

Elaborado por Wendy German y Dorival Pichamba

Figura 22*Diagrama conceptual banco vacuna*

Elaborado por Wendy German y Dorival Pichamba

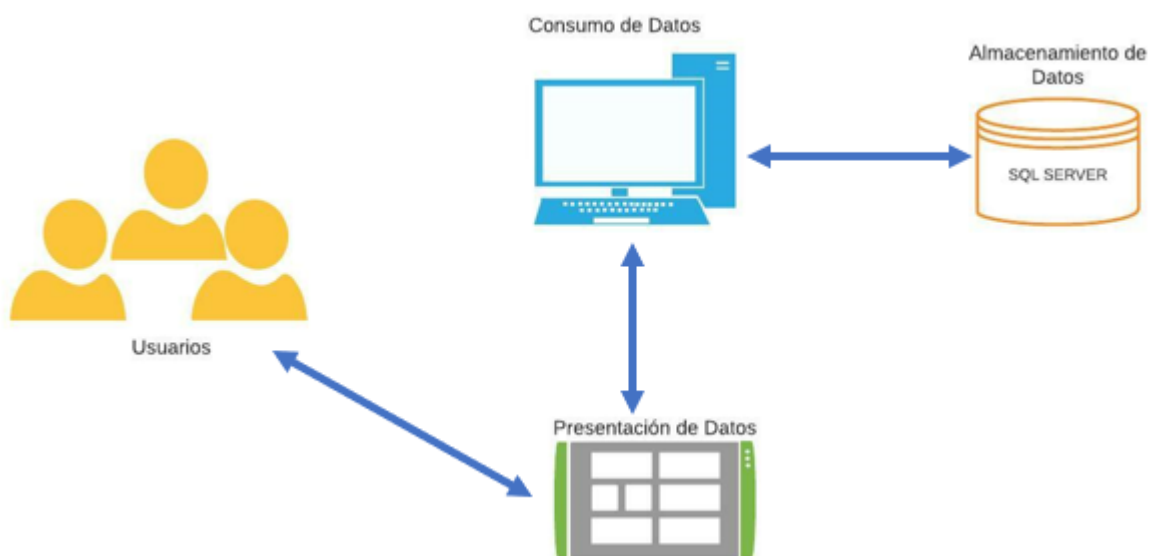
CAPÍTULO 3 CONSTRUCCIÓN Y PRUEBAS

3.1 Arquitectura Propuesta

Este proyecto se construyó bajo una arquitectura basada en MVC (Modelo – Vista - Controlador), el cual maneja gran cantidad de datos y transacciones complejas facilitando los procesos y haciendo al sistema más seguro y con respuestas inmediatas.

Figura 23

Modelo MCV utilizado en el Sistema

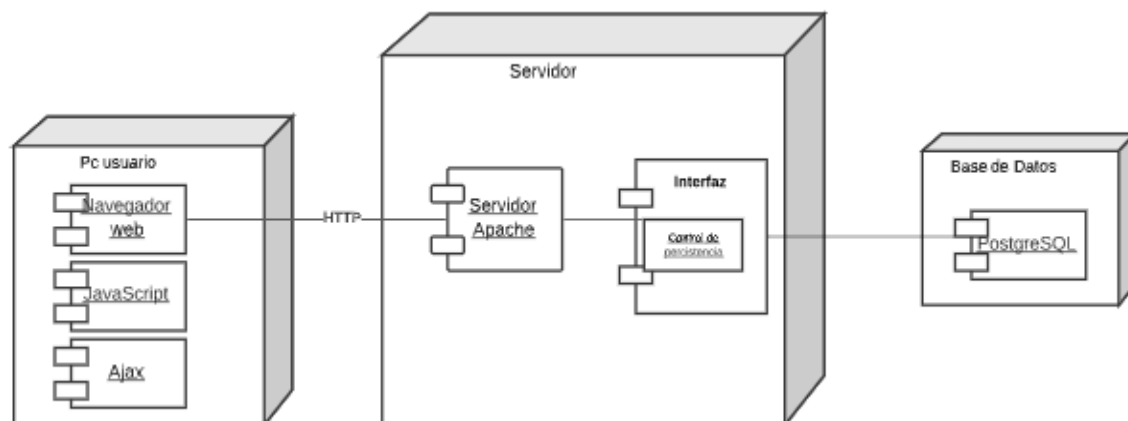


Elaborado por Wendy German y Dorival Pichamba

Como muestra la figura 20 el sistema muestra al usuario logueado y la información se encuentra en un almacenamiento de datos para que sean gestionados según el rol designado.

3.2 Diagrama de Despliegue

En la figura 24 se muestra el diagrama de despliegue que representa la estructura y las relaciones físicas de los distintos nodos que componen el sistema.

Figura 24*Diagrama de despliegue*

Elaborado por Wendy German y Dorival Pichamba

3.3 Diagrama Físico de la Base de Datos

Figura 25

Diagrama físico de la Base de Datos



Elaborado por Wendy German y Dorival Pichamba

3.4 Métodos y Algoritmos Generados

A continuación, se muestran los algoritmos y métodos más importantes utilizados en el sistema informático de vacunas:

Módulo de Bodega:

El código recibe como parámetros el idfecha y el idbiologico que se selecciona en los campos para luego realizar una consulta a la base de datos para verificar si existe un valor anterior al id_fecha actual y del mismo id_biologico en la columna desperdicio para ubicarlo en el campo saldo anterior del formulario actual.

Figura 26

Código para consultar la cantidad sobrante del mes anterior

```
$idfecha={id_fecha}-1;
$idbio={id_biologico};

$sql="SELECT total_disponible FROM kardex WHERE id_biologico=$idbio and id_fecha=$idfecha";
sc_lookup(ds,$sql);
$sobrante={ds[0][0]};

{saldo_anterior}=$sobrante;
```

Elaborado por Wendy German y Dorival Pichamaba

Método que se utiliza para sumar los campos donde se recibe la dosis de un saldo anterior y la cantidad recibida del mes actual.

Figura 27

Código para sumar la dosis del saldo anterior y la cantidad recibida del mes actual

```
{total}={cantidad_recibida}+{saldo_anterior};
```

Elaborado por Wendy German y Dorival Pichamba

Método que se utiliza para registrar en un reporte Kardex, las vacunas ingresadas cada mes donde recibe datos como el id_fecha que ingresa, el establecimiento (unicodigo), la cantidad recibida, saldo anterior y el total calculado.

Figura 28

Código para registrar las vacunas ingresadas en el reporte Kardex

```
$idfecha={id_fecha};
$idbio={id_biologico};

$unicodigo={unicodigo};
$cantrec={cantidad_recibida};
$saldoant={saldo_anterior};
$total={total};

$sql="INSERT INTO kardex (id_fecha,unicodigo,id_biologico,cantidad_recibida,
saldo_anterior,total,total_disponible,dosis_administrada,desperdicio)
VALUES ($idfecha,$unicodigo, $idbio,$cantrec,
$saldoant, $total, 0, 0, 0)";
sc_exec_sql($sql);
sc_commit_trans();
```

Elaborado por Wendy German y Dorival Pichamba

Método mSolicitud que se utiliza en un evento ajax idBiologicoOnChange para que al elegir en el campo select una de las opciones, se consulte de la tabla kardex los datos sobre saldo anterior, cantidad recibida, dosis administrada, desperdicios con los parámetros de id_biologico e id_fecha:

Figura 29

Código para elegir opciones y consultar datos sobre el saldo anterior y cantidad recibida

```
$idbio={id_biológico};
$idfechater={id_fecha_termino};

$sql="SELECT saldo_anterior, cantidad_recibida,
dosis_administrada, desperdicio FROM kardex
WHERE id_biológico=$idbio AND id_fecha=$idfechater" ;
sc_lookup(ds,$sql);
$saldoant={ds[0][0]};
$cantrecb={ds[0][1]};
$dosis_administrada={ds[0][2]};
$desp={ds[0][3]};
{saldo_anterior}=$saldoant;
{del_banco_de_vacunas}=$cantrecb;
{aplicadas_en_mes}=$dosis_administrada;
{perdida_vacuna_fco_abierto}=$desp;
```

Elaborado por Wendy German y Dorival Pichamba

Módulo de Enfermería:

Evento onLoad para que al cargar el formulario se pueda visualizar los campos de identificación donde consiste que el identificador si es número 2 se visualiza el campo cédula, caso contrario en número 3 se visualiza el campo pasaporte y finalmente el identificador 4 pertenece al campo de la generación del código de 17 dígitos.

Figura 30

Código para clasificar el tipo de identificación

```
$identificacion={id_identificacion};

if($identificacion==2){
    sc_field_display({cedula_pasaporte},on);
    sc_field_display({sin_identificacion},off);
}else if($identificacion==3){
    sc_field_display({cedula_pasaporte},on);
    sc_field_display({sin_identificacion},off);
}else if($identificacion==4){
    sc_field_display({sin_identificacion},on);
    sc_field_display({cedula_pasaporte},off);
}else{
    sc_field_display({sin_identificacion},off);
    sc_field_display({cedula_pasaporte},off);
}
```

Elaborado por Wendy German y Dorival Pichamba

Método para realizar el proceso de generación de código de 17 dígitos para las personas sin identificación, consiste en tomar 2 letras del primer nombre, 1 letra del segundo nombre 2 letras del primer apellido, 1 letra del segundo apellido, código de provincia de donde nació, datos de la fecha de nacimiento y la década del año en que nació.

Figura 31

Código para generar el código de 17 dígitos

```
$identificacion={id_identificacion};
$primernombre={primer_nombre};
$segundonombre={segundo_nombre};
$primerapellido={primer_apellido};
$segundoapellido={segundo_apellido};
$provincia={lugar_nacimiento};
$fechanac={fecha_nacimiento};
$anios=date("Y",strtotime($fechanac));
$decada=str_split($anios,1);
$mess = date("m", strtotime($fechanac));
$dias = date("d", strtotime($fechanac));
$nombre=str_split($primernombre,2);
$nombre2=str_split($segundonombre,1);
$apellido=str_split($primerapellido,2);
$apellido2=str_split($segundoapellido,1);
$indocumentado=array($nombre[0],$nombre2[0],$apellido[0],$apellido2[0],$provincia,$anios,$mess,$dias,$decada[2]);
```

```
if($identificacion==2){
    //sc_alert("HOLA");
    sc_field_display({cedula_pasaporte},on);
    sc_field_display({sin_identificacion},off);
}else if($identificacion==3){
    sc_field_display({cedula_pasaporte},on);
    sc_field_display({sin_identificacion},off);
}else if($identificacion==4){
    sc_field_display({sin_identificacion},on);
    sc_field_display({cedula_pasaporte},off);
    {sin_identificacion}=implode("", $indocumentado);
}else{
    sc_field_display({sin_identificacion},off);
    sc_field_display({cedula_pasaporte},off);
}
```

Elaborado por Wendy German y Dorival Pichamba

Algoritmo para insertar datos en caso de seleccionar la captación temprana donde recibirá datos de la vacuna y sus dosis respectivas.

Figura 32

Código para insertar datos en captación temprana

```

if({tipo_captacion_tem}=="TEMPRANA"){

    $cap1={tipo_captacion_tem};
    echo "Ha seleccionado $cap1";

    //$bcg={pdtem_vac_bcg};//aqui
    $dosbcg={pdtem_dos_bcg};

    //$hbped={pdtem_vac_hbped};
    $doshbped={pdtem_dos_hbped};

    //$rotarix={pdtem_vac_rotarix};
    $dos1rotarix={pdtem_dos1_rotarix};
    $dos2rotarix={pdtem_dos2_rotarix};

    //$ipv={pdtem_vac_ipv};
    $dos1ipv={pdtem_dos1_ipv};
    $dos2ipv={pdtem_dos2_ipv};

    $Neumococodos1={pdtem_dos1_neumococo};
    $Neumococodos2={pdtem_dos2_neumococo};
    $Neumococodos3={pdtem_dos3_neumococo};

    $Pentavalentedos1={pdtem_dos1_pentavalente};
    $Pentavalentedos2={pdtem_dos2_pentavalente};
    $Pentavalentedos3={pdtem_dos3_pentavalente};

    $bopvdos2={pdtem_dos2_bopv};
    $bopvdos3={pdtem_dos3_bopv};
    $bopvref1={pdtem_ref1_bopv};
    $bopvref2={pdtem_ref2_bopv};

    $srpdos1={pdtem_dos1_srp};
    $srpdos2={pdtem_dos2_srp};

    $fados={pdtem_dos_fa};
    $varicelados={pdtem_dos_varicela};

    $dptref1={pdtem_ref1_dpt};
    $dptref2={pdtem_ref2_dpt};

    $hpvdos1={pdtem_dos1_hpv};
    $hpvdos2={pdtem_dos2_hpv};

    $dtAdultotemref3={pdtem_ref3_dtadulto};

```

Elaborado por Wendy German y Dorival Pichamba

En este método se inserta los valores de dosis solo para la captación temprana.

Figura 33

Código para insertar la dosis en captación temprana

```
$dtAdultotemref3={pdtem_ref3_dtadulto};

$sql1 ="INSERT INTO captacion_temprana(id_paciente,id_fecha,
pdtem_vac_bcg, pdtem_dos_bcg,
pdtem_vac_hbped, pdtem_dos_hbped,
pdtem_vac_rotarix,pdtem_dos1_rotarix,pdtem_dos2_rotarix,
pdtem_vac_ipv,pdtem_dos1_ipv,pdtem_dos2_ipv,
pdtem_vac_neumococo,pdtem_dos1_neumococo,pdtem_dos2_neumococo,pdtem_dos3_neumococo,
pdtem_vac_pentavalente,pdtem_dos1_pentavalente,pdtem_dos2_pentavalente,pdtem_dos3_pentavalente,
pdtem_vac_bopv,pdtem_dos2_bopv,pdtem_dos3_bopv,pdtem_ref1_bopv,pdtem_ref2_bopv,
pdtem_vac_srp,pdtem_dos1_srp,pdtem_dos2_srp,
pdtem_vac_fa,pdtem_dos_fa,
pdtem_vac_varicela,pdtem_dos_varicela,
pdtem_vac_dpt,pdtem_ref1_dpt,pdtem_ref2_dpt,
pdtem_vac_hpv,pdtem_dos1_hpv,pdtem_dos2_hpv,
pdtem_vac_dtadulto,pdtem_ref3_dtadulto
) VALUES ($idpac, $idfecha,
'BCG', $dosbcg,
'HB Pediátrica', $doshbped,
'ROTARIX', $dos1rotarix, $dos2rotarix,
'IPV', $dos1ipv, $dos2ipv,
'NEUMOCOCO DECAVALENTE', $Neumococodos1, $Neumococodos2, $Neumococodos3,
'PENTAVALENTE /T', $Pentavalentedos1, $Pentavalentedos2, $Pentavalentedos3,
'POLIO BIVALENTE', $bopvdos2, $bopvdos3, $bopvref1, $bopvref2,
'SRP(UNIDOSIS)', $srpdos1, $srpdos2,
'FA', $fados,
'VARICELA', $varicelados,
'DPT', $dptref1, $dptref2,
'HPV', $hpvdos1, $hpvdos2,
'dT. ADULTO', $dtAdultotemref3)";
sc_exec_sql($sql1);
sc_commit_trans();
```

Elaborado por Wendy German y Dorival Pichamba

En el presente código si se selecciona la dosis respectiva se consultará la cantidad disponible de dicha vacuna y se le restará uno al guardar el registro, además se consulta que cantidad de dosis han sido administradas para finalmente realizar una actualización en las tablas de kardex y configuración del biológico donde se registra los datos de las dosis disponibles que se hayan ingresado.

Figura 34

Código para seleccionar la dosis de la vacuna administrada

```
$temprano={tipo_captacion_tem};
if ($temprano == "TEMPRANA"){
    sc_alert("DIRIGIRSE A LA PESTAÑA CAPTACION $temprano");
    sc_block_display(bl_temprana, on);
    $anio={cp_anio_paciente};
    $mes={cp_mes_paciente};
    $dia={cp_dia_paciente};

    if($anio==0 AND $mes==0 AND $dia==0){
        m_24primerashoras();
    }else if($anio==0 AND ($mes>=0 AND $mes<=11) AND ($dia>=1 AND $dia<=29)){
        m_meses111();
    }else if($anio==1 AND ($mes>=0 AND $mes<=11) AND ($dia>=1 AND $dia<=29)){
        m_11_23meses();
    }else if($anio==5 AND ($mes>=0 AND $mes<=11) AND ($dia>=1 AND $dia<=29)){
        m_5anios();
    }else if($anio==9 AND ($mes>=0 AND $mes<=11) AND ($dia>=1 AND $dia<=29)){
        m_9anios();
    }else if($anio==15 AND ($mes>=0 AND $mes<=11) AND ($dia>=1 AND $dia<=29)){
        m_15anios();
    }
}
```

Elaborado por Wendy German y Dorival Pichamba

Método tipo captación tardía_onClick donde el usuario al elegir la opción “TARDÍO” despliega los bloques de vacunas según los parámetros de edad del esquema de vacunación para dicha captación.

Figura 35

Código para desplegar los bloques de vacunas

```
$tardio={tipo_captacion_tar};
if ($tardio == "TARDÍO"){
    sc_alert("DIRIGIRSE A LA PESTAÑA CAPTACION $tardio ");
    sc_block_display(bl_tardía_vacunas, on);

    $anio={cp2_anio_paciente};
    $mes={cp2_mes_paciente};
    $dia={cp2_dia_paciente};

    if($anio==1 AND ($mes>=0 AND $mes<=11) AND ($dia>=1 AND $dia<=29)){
        mTar1anio();
    }else if(($anio>=2 AND $anio<=5) AND ($mes>=0 AND $mes<=11) AND ($dia>=1 AND $dia<=29)){
        mTar2anio();
    }else if(($anio>=7 AND $anio<=8) AND ($mes>=0 AND $mes<=11) AND ($dia>=1 AND $dia<=29)){
        mTar67anio();
    }else if(($anio>=51 AND $anio<=59) AND ($mes>=0 AND $mes<=11) AND ($dia>=1 AND $dia<=29)){
        mTar859anio();
    }else if(($anio>=8 AND $anio<=50) AND ($mes>=0 AND $mes<=11) AND ($dia>=1 AND $dia<=29)){
        mTar850anio();
    }
}
```

Elaborado por Wendy German y Dorival Pichamba

Método tipo captación dt_onClick donde el usuario al elegir la opción "Dt. PEDIATRICO" despliega los bloques de vacunas según los parámetros de edad del esquema de vacunación para dicha captación.

Figura 36

Código para desplegar los bloques de vacunas según los parámetros de edad del paciente

```
$pediatrica={tipo_captacion_dt};

if ($pediatrica == "Dt. PEDIATRICO"){
    sc_alert("DIRIGIRSE A LA PESTAÑA DE CAPTACION $pediatrica");
    sc_block_display(bl_dt_pediatrica, on);

    $anio={cp3_anio_paciente};
    $mes={cp3_mes_paciente};
    $dia={cp3_dia_paciente};

    if(($anio>=7 AND $anio<=14)AND($mes>=0 AND $mes<=11) AND ($dia>=1 AND $dia<=29)){
        mDTadulto();
    }

    if($anio==6 AND($mes>=0 AND $mes<=11) AND ($dia>=1 AND $dia<=29)){
        mDTnino();
    }
}
```

Elaborado por Wendy German y Dorival Pichamba

Método aplicado en todas las dosis de vacunas donde consulta si existe dosis disponibles para dicho id_biológico para desplegar las dosis correspondientes caso contrario emitirá un error y el usuario no tendrá acceso a los campos de dosis para administrar.

Figura 37

Código general para consultar las dosis disponibles

```
$fecha_atencion={fecha_atencion};
$unicodigo={unicodigo};
$idfecha={id_fecha};
$bio="SELECT total FROM configuracion_biologico WHERE id_fecha=$idfecha AND id_biologico=11 AND unicodigo=$unicodigo";
sc_lookup(vac,$bio);
$TOTAL_VACUNA={vac[0][0]};

if ($TOTAL_VACUNA > 0){

    sc_block_display(bl_dosis_pdadulto,on);

    sc_field_display(pddt_dos1_dtadulto,on);
    sc_field_display(pddt_dos2_dtadulto,on);
    sc_field_display(pddt_dos3_dtadulto,on);
    sc_field_display(pddt_dos4_dtadulto,on);
    sc_field_display(pddt_dos5_dtadulto,on);
}else{

    sc_alert("VACUNA AGOTADAS o INEXISTENTES");

}
```

Elaborado por Wendy German y Dorival Pichamba

Método para calcular el año, mes y día del paciente al ingresar la fecha de nacimiento y la fecha en que se atendió.

Figura 38

Código para calcular la edad del paciente

```
{resultado} = sc_date_dif_2({fecha_nacimiento},
"yyyy-mm-dd", {fecha_atencion}, "yyyy-mm-dd",1);

{cp_dia_paciente} = {resultado[0]};
{cp_mes_paciente} = {resultado[1]};
{cp_anio_paciente} = {resultado[2]};

{cp2_dia_paciente} = {resultado[0]};
{cp2_mes_paciente} = {resultado[1]};
{cp2_anio_paciente} = {resultado[2]};

{cp3_dia_paciente} = {resultado[0]};
{cp3_mes_paciente} = {resultado[1]};
{cp3_anio_paciente} = {resultado[2]};
```

Elaborado por Wendy German y Dorival Pichamba

Algoritmo utilizado en Kardex diario para actualizar las cantidades disponibles de dosis en caso de haber desperdicios en ciertas vacunas.

Figura 39

Código para actualizar las cantidades disponibles si hubo desperdicios

```
$idbio={id_biológico};
$idfecha={id_fecha};
$unicodigo={unicódigo};
$desp={desperdicio};
$total={total};
//*****Actualizar recepcion de vacunas
$sql="UPDATE configuracion_biológico SET total = $total WHERE id_fecha= $idfecha AND id_biológico = $idbio and unicódigo=$unicódigo";
sc_exec_sql($sql);
sc_commit_trans();
//*****Actualizar campos de kardex general
$sql2="UPDATE kardex SET total_disponible=$total WHERE id_fecha= $idfecha AND id_biológico = $idbio and unicódigo=$unicódigo";
sc_exec_sql($sql2);
sc_commit_trans();
//*****Consulta desperdicio
$sql3="SELECT desperdicio from kardex WHERE id_fecha=$idfecha and id_biológico=$idbio and unicódigo=$unicódigo";
sc_lookup(ds,$sql3);
$desper={ds[0][0]};
//$cont=$desper+1;
$conr=$desper+$desp;
//sc_alert("DESPERDICIOS $conr");
$sql4="UPDATE kardex SET desperdicio=$conr WHERE id_fecha= $idfecha AND id_biológico = $idbio and unicódigo=$unicódigo";
sc_exec_sql($sql4);
sc_commit_trans();
```

Elaborado por Wendy German y Dorival Pichamba

Método de Kardex diario para validar de que vacunas se obtiene un desperdicio y para consultar las dosis administradas y cantidades disponibles en tiempo real.

Figura 40

Código para validar que vacunas tienen desperdicios y consultar las dosis

```
$idbio={id_biologico};
$idfecha={id_fecha};
if($idbio==1 OR $idbio ==4 OR $idbio==7 OR $idbio==8
OR $idbio==11 OR $idbio==14 OR $idbio==15 OR $idbio==16){
    sc_field_display({desperdicio},on);
}else{
    sc_field_display({desperdicio},off);
}
$sql="SELECT cantidad_recibida, saldo_anterior,total
FROM configuracion_biologico
WHERE id_fecha=$idfecha and id_biologico=$idbio";
sc_lookup(ds,$sql);
$recibidacant={ds[0][0]};
$anteriosaldo={ds[0][1]};
$totalvac={ds[0][2]};

{saldo_anterior}=$anteriosaldo;
{cantidad_recibida}=$recibidacant;
{total}=$totalvac;
```

Elaborado por Wendy German y Dorival Pichamba

Método unicodigoOnChange para consultar el consolidado mensual de la captación temprana con tan solo elegir el establecimiento donde se realizó la administración de vacunas.

Figura 41

Código para consultar el consolidado mensual de la captación temprana

```

$unicodigo={unicodigo};
$idfechater={id_fecha_termino};
establecimiento();
sexo();
nacionalidad();
etnico();
$var=1;

$bcg="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_vac_bcg='BCG'";
sc_lookup(ds1,$bcg);
{pdtem_vac_bcg}={ds1[0][0]};

$dosu="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_dos_bcg=$var";
sc_lookup(ds2,$dosu);
{pdtem_dos_bcg}={ds2[0][0]};

$hbped="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_vac_hbped='HB Pediátrica'";
sc_lookup(ds3,$hbped);
{pdtem_vac_hbped}={ds3[0][0]};

$dosu2="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_dos_hbped=$var";
sc_lookup(ds4,$dosu2);

$rotarix="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_vac_rotarix='ROTARIX'";
sc_lookup(ds5,$rotarix);
{pdtem_vac_rotarix}={ds5[0][0]};

$primrot="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_dos1_rotarix=$var";
sc_lookup(ds6,$primrot);
{pdtem_dos1_rotarix}={ds6[0][0]};

$segrot="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_dos2_rotarix=$var";
sc_lookup(ds7,$segrot);
{pdtem_dos2_rotarix}={ds7[0][0]};

$ipv="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_vac_ipv='IPV'";
sc_lookup(ds8,$ipv);
{pdtem_vac_ipv}={ds8[0][0]};

$primipv="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_dos1_ipv=$var";
sc_lookup(ds9,$primipv);
{pdtem_dos1_ipv}={ds9[0][0]};

$segipv="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_dos2_ipv=$var";
sc_lookup(ds10,$segipv);
{pdtem_dos2_ipv}={ds10[0][0]};

$neumococo="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_vac_neumococo='NEUMOCOCO DECAVALENTE'";
sc_lookup(ds11,$segipv);
{pdtem_vac_neumococo}={ds11[0][0]};

$dos1neumo="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_dos1_neumococo=$var";
sc_lookup(ds12,$dos1neumo);
{pdtem_dos1_neumococo}={ds12[0][0]};

$dos2neumo="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_dos2_neumococo=$var";
sc_lookup(ds13,$dos2neumo);
{pdtem_dos2_neumococo}={ds13[0][0]};

$dos3neumo="SELECT count(*) FROM paciente WHERE id_fecha=$idfechater and unicodigo=$unicodigo AND pdtem_dos3_neumococo=$var";
sc_lookup(ds14,$dos3neumo);
{pdtem_dos3_neumococo}={ds14[0][0]};

```

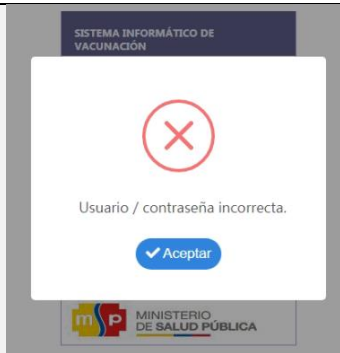
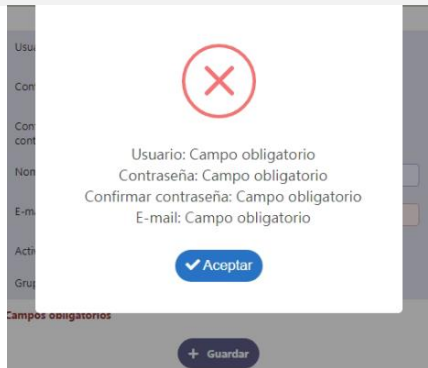
Elaborado por Wendy German y Dorival Pichamba

3.5 Pruebas

3.5.1 Caja Negra

Tabla 12

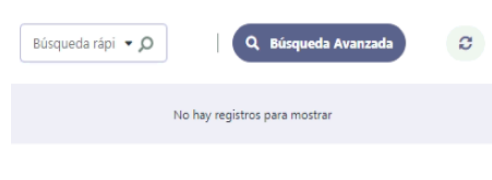
Resultados de las pruebas de caja negra del sistema

Requerimiento Funcional	Flujo Alternativo	Mensaje de error
RF1	El sistema autentifica al usuario y en caso de ser incorrecto genera un mensaje de error	
RF2	En caso de dejar un campo vacío genera errores de campo obligatorio.	
RF3	En caso de existir un campo vacío genera el error de campo obligatorio.	

En caso de no existir registros emite el error de registros inexistentes.

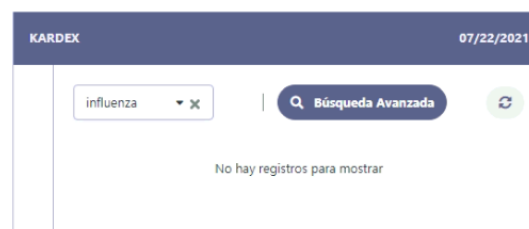


En caso de no existir registros emitirá el mensaje de registros inexistentes.



RF4

Al realizar la filtración de un registro inexistente aparece un error de registros inexistentes.

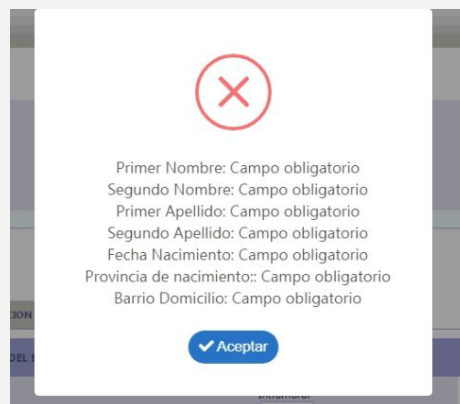


En caso de no existir registros emite el error de registro inexistente

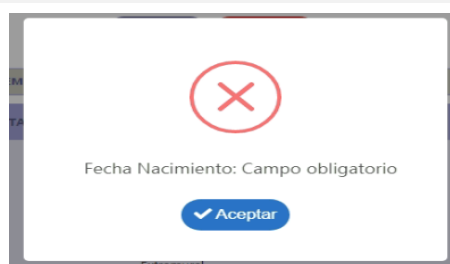


RF5

En caso de dejar vacío algún campo del registro de pacientes, emitirá el error de campo obligatorio y no podrá guardar el registro.



En caso de no llenar el campo de fecha de nacimiento, emitirá el error de campo obligatorio.



RF6

En caso de no seleccionar una captación no mostrará ninguna vacuna en las captaciones por campo obligatorio.

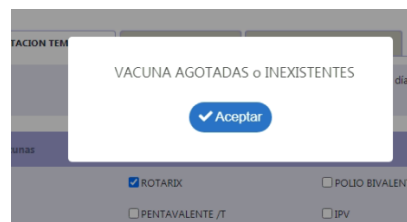
 A screenshot of a web form. At the top are three tabs: 'Registro de Pacientes' (selected), 'CAPTACION TEMPRANA', and 'CAPTACION TARDIA'. Below the tabs are two input fields: 'Edad del paciente en años:' with a dropdown showing '0', and 'Edad del paciente en meses:' with a dropdown showing '5'. Below these is a red asterisk followed by the text '* Campos obligatorios'.

En caso de no seleccionar alguna vacuna de la captación, no desplegará las dosis

 A screenshot of the same web form as above, but with the 'CAPTACION TEMPRANA' tab selected. Below the age fields, there is a section titled 'CAPTACION TEMPRANA: Vacunas' with a dropdown arrow. Under this section are two checkboxes: 'BCG' and 'NEUMOCOCO DECAVALENTE'. Below these is a red asterisk followed by the text '* Campos obligatorios'.

respectivas por campo obligatorio.

En caso de que las vacunas se agoten, emitirá un mensaje de error de vacunas agotadas o inexistentes por lo que no podrá acceder a las dosis respectivas.



Al no existir registros en las captaciones emiten el error de no hay registros para mostrar.



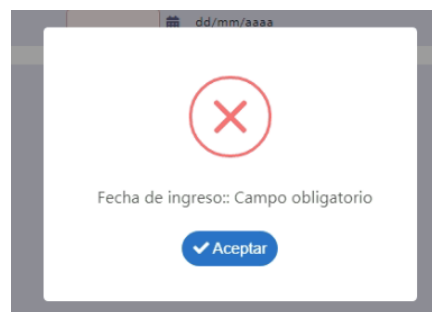
RF7

Al realizar una búsqueda y no existe el registro emite el siguiente mensaje, no hay registros para mostrar.



RF8

El sistema al detectar
el campo fecha vacío
emite el error de
campo obligatorio



El sistema valida los
campos vacíos y
emite el erro de
“campos obligatorios”
que son requeridos
para el registro



En caso no existir
registro emitirá el
mensaje “No hay
registros para
mostrar”



3.5.2 Carga

Para realizar las pruebas de carga del sistema informático de vacuas se ha utilizado la herramienta de Apache JMeter donde se ha utilizado todos los procesos existentes en el programa con un número aproximado de usuarios que se observa en la columna “# Muestras” concurrentes que utilizará los módulos por minuto. Se puede visualizar los encabezados de las columnas que tienen la siguiente estructura:

- Label: Nombre de etiqueta dado a la respectiva solicitud HTTP
- #Samples: El valor de la cantidad de usuarios por solicitud
- Average: El tiempo promedio que tarda en ejecutar una etiqueta específica
- Min: El tiempo mínimo de respuesta que tiene las muestras en una etiqueta
- Max: El tiempo máximo de respuesta que tiene las muestras en una etiqueta
- Std. Dev: Muestra el conjunto de casos excepcionales que se desvían del valor medio del tiempo de respuesta de la muestra.
- Error: Porcentaje de error en resputas fallidas de las etiquetas
- Throughput: Es el rendimiento que procesa al ejecutar las peticiones en las etiquetas
- Received Kb/sec: Es la cantidad de datos que recibe o descarga la etiqueta
- Sent Kb/sec: Es la cantidad de datos que envía la etiqueta
- Avg Bytes: Un promedio de datos de la cantidad de datos descargados del servidor durante la ejecución.

Tabla 13

Resultados de las pruebas de carga del sistema

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
VacunasLogin	400	22979	2407	221902	24117.39	1.25%	1.8/sec	294.91	0.39	168678.6
VacunasMenu	395	146	0	13581	809.41	34.18%	9.5/sec	18.83	0.83	2035.6
TOTAL	795	11635	0	221902	20574.61	17.61%	3.6/sec	298.42	0.55	85881.2

Elaborado por la herramienta JMeter – Summary Report

En las siguientes tablas se muestra los resultados que simula la herramienta de una cantidad determinada de usuarios que realizan pedidos al servidor donde se encuentra alojado el sistema haciendo uso de las diferentes etiquetas.

Tabla 14

Resultados del test de carga con número mínimo de usuarios

Etiqueta	# Muestras	Método	Tiempo Medio (ms)	Tiempo Min (ms)	Tiempo Max (ms)	% Error	Rendimiento	Kb/sec	Sent KB/sec
VacunasLogin	18	POST	429	307	538	0,000 %	18,9	52,71	0,09
VacunasMenu	18	GET	195	137	290	0,000 %	19,0	0,65	0,06
VacunasFormulariosFechaEntrega	6	POST	614	134	2807	0,000 %	7,2	0,25	0,03
VacunasFormularioRecepcion	6	GET	167	119	248	0,000 %	7,2	0,25	0,03
VacunasFormulariosFechaEntregaPaciente	5	GET	102	81	121	0,000 %	6,2	0,22	0,02
VacunasFormularioPaciente	5	POST	142	101	171	0,000 %	6,2	0,22	0,03
VacunasReporteCapTem	6	GET	161	88	212	0,000 %	7,2	0,25	0,02
VacunasReporteCapTar	6	GET	136	92	237	0,000 %	7,2	0,25	0,02

VacunasReporteCapDt	6	GET	168	89	285	0,000 %	7,1	0,25	0,02
VacunasReporteKardexMensual	6	GET	110	86	168	0,000 %	7,2	0,25	0,02
VacunasFormulariosFechaEntregaKardex	4	GET	107	70	174	0,000 %	5,3	0,19	0,02
VacunasFormulariosKardexDiario	4	POST	149	126	165	0,000 %	5,3	0,19	0,02
VacunasReporteKardexDiario	5	GET	224	115	386	0,000 %	6,2	0,22	0,02
VacunasFormulariosFechaTermino	6	POST	127	80	160	0,000 %	7,2	0,25	0,03
VacunasFormulariosSolicitud	6	POST	94	76	140	0,000 %	7,2	0,25	0,03
VacunasReporteSolicitudVacuna	6	GET	156	104	225	0,000 %	7,2	0,25	0,02
VacunasFormulariosFechaTerminoConsolidadoTem	5	GET	123	87	158	0,000 %	6,2	0,22	0,02
VacunasFormularioCTemcoconsolidado	5	POST	141	82	206	0,000 %	6,2	0,22	0,03
VacunasFormulariosFechaTerminoConsolidadoTardio	5	GET	84	64	113	0,000 %	6,2	0,22	0,02
VacunasFormularioCTarconsolidado	5	POST	130	88	189	0,000 %	6,3	0,22	0,03
VacunasFormulariosFechaTerminoConsolidadoDt	5	GET	115	69	155	0,000 %	6,3	0,22	0,02
VacunasFormularioCDtconsolidado	5	POST	130	100	148	0,000 %	6,3	0,22	0,03
VacunasReporteConsolidadoTem	5	GET	125	82	187	0,000 %	6,2	0,22	0,02
VacunasReporteConsolidadoTar	5	GET	109	72	169	0,000 %	6,2	0,22	0,02

VacunasReporteConsolidadoDt	5	GET	112	87	127	0,000 %	6,2	0,22	0,02
Promedio	6.32		166	101.44	311.16	0	7.516	2.34	0.0276

Elaborado por Wendy German y Dorival Pichamba

Para observar el comportamiento del sistema se procederá a duplicar sus valores de usuarios concurrentes que realicen peticiones por minuto al sistema informático en todos los módulos existentes.

Tabla 15

Resultados del test de carga con el doble de usuarios

Etiqueta	# Muestras	Método	T. Media (ms)	T. Min (ms)	T. Max (ms)	% Error	Rendimiento	Kb/sec	Sent KB/sec
VacunasLogin	36	POST	449	293	805	0,00 0%	35.8	99.82	0.18
VacunasMenu	36	GET	195	136	309	0,00 0%	36.2	1.23	0.12
VacunasFormulariosFechaEntrega	12	POST	126	75	179	0,00 0%	12	0.42	0.06
VacunasFormularioRecepcion	12	GET	116	67	157	0,00 0%	12	0.42	0.06
VacunasFormulariosFechaEntrega Paciente	10	GET	88	70	144	0,00 0%	10	0.35	0.03
VacunasFormularioPaciente	10	POST	156	118	241	0,00 0%	10	0.35	0.04
VacunasReporteCapTem	12	GET	159	104	232	0,00 0%	12	0.42	0.04
VacunasReporteCapTar	12	GET	173	99	263	0,00 0%	12	0.42	0.04
VacunasReporteCapDt	12	GET	144	86	288	0,00 0%	12	0.42	0.04
VacunasReporteKardexMensual	12	GET	146	88	232	0,00 0%	12	0.42	0.04
VacunasFormulariosFechaEntrega Kardex	8	POST	111	69	178	0,00 0%	8	0.28	0.03

VacunasFormulariosKardexDiario	8	POST	111	71	14 5	0,00 0%	8	0.28	0.03
VacunasReporteKardexDiario	10	GET	162	10 1	20 1	0,00 0%	10	0.35	0.03
VacunasFormulariosFechaTermino	12	POST	122	72	17 1	0,00 0%	12	0.42	0.06
VacunasFormulariosSolicitud	12	POST	120	72	16 8	0,00 0%	12	0.42	0.05
VacunasReporteSolicitudVacuna	12	GET	144	72	21 7	0,00 0%	12	0.42	0.04
VacunasFormulariosFechaTermino ConsolidadoTem	10	GET	102	68	15 2	0,00 0%	10	0.35	0.03
VacunasFormularioCTemconsolidado	10	POST	114	92	18 2	0,00 0%	10	0.35	0.05
VacunasFormulariosFechaTermino ConsolidadoTardio	10	GET	93	64	14 0	0,00 0%	10	0.35	0.03
VacunasFormularioCTarconsolidado	10	POST	136	84	17 9	0,00 0%	10	0.35	0.05
VacunasFormulariosFechaTermino ConsolidadoDt	10	GET	112	64	15 4	0,00 0%	10	0.35	0.03
VacunasFormularioCDtconsolidado	10	POST	114	78	15 9	0,00 0%	10	0.35	0.05
VacunasReporteConsolidadoTem	10	GET	138	75	20 4	0,00 0%	10	0.35	0.03
VacunasReporteConsolidadoTar	10	GET	144	86	22 9	0,00 0%	10	0.35	0.03
VacunasReporteConsolidadoDt	10	GET	128	90	19 8	0,00 0%	10	0.35	0.03
Promedio	12.64		144 .1	91 .8	22 1.1	0,0 %	12.64	4.38	0.04

Elaborado por Wendy German y Dorival Pichamba

3.5.3 Stress

Para las pruebas de estrés utilizamos la herramienta de Apache JMeter, donde se dará uso al Login porque es una sección donde se pueden presentar la mayor cantidad de usuarios concurrentes en segundos. La prueba consistió en enviar 300 peticiones al servidor realizadas

en 1 segundo para luego aumentar el número de muestras con el objetivo de analizar el funcionamiento del sistema informático de vacunas.

Tabla 16

Resultados del test de stress del sistema

Etiqueta	# Muestras	Método	T. Medio (ms)	T. Min (ms)	T. Max (ms)	% Error	Rendimiento	Received Kb/sec	Sent KB/seg
VacunasLogin	300	POST	71285	1361	141501	0.00 %	2.1/sec	352.95	0.63
	400	POST	77727	1597	1706176	1%	14.1/min	38.71	0.05
	450	POST	38099	995	2421399	3.11 %	11.1/min	29.99	0.04

Elaborado por Wendy German y Dorival Pichamba

3.6 Diagnóstico de Resultados

Test de carga

La prueba de carga realizado en la Tabla 14 muestra una media de 6.32, el tiempo medio de respuesta es de 0.166 segundos por cada petición, su tiempo mínimo es de 0.1014 segundos y el máximo es de 0.31116. En base a los resultados se puede ver que el promedio en rendimiento es de 7.516 solicitudes por segundo, la cantidad de descargada del servidor durante su ejecución fue de 2.34kb por segundo y los enviados fueron 0.0276Kb/s. Al analizar la columna de Error en todos los procesos se puede notar que no existe un porcentaje mayor del 0% de manera que el tiempo de respuesta para el usuario es aceptable.

Con los resultados obtenidos en la Tabla 15, la media de muestras es 12.64 en un minuto, el tiempo promedio es de 0.11412 segundos por cada petición, su tiempo mínimo tiene un valor de 0.09176 y con un máximo de 0.22108. Asimismo, el rendimiento es de 12.64 en solicitudes, la cantidad descarga del servidores durante la ejecución fue de 4.38 y los datos

enviados son de 0.04KB/s. Al analizar las dos tablas se observa que, al duplicar el número de peticiones enviadas, el valor de rendimiento se ha igualado al número promedio de muestras enviadas, por lo que el sistema no soportaría triplicar la cantidad de usuarios con sus peticiones al servidor ya que si su número aumenta su rendimiento tiende a disminuir y tener una respuesta lenta de servidor.

Prueba de Stress

Según los resultados de la Tabla 16 se ha realizado un número de peticiones con valor de 300 a lo que el sistema responde bien con error del 0% con un tiempo promedio de respuesta de 71.285 que es aceptable, pero a medida que aumenta de peticiones empieza a generar errores de TimeOut y el tiempo de respuesta empieza a disminuir en la parte de Login y para experiencia de usuario es algo ineficiente si el sistema aumenta más de 400 peticiones al servidor.

Conclusiones

- La recopilación de reportes en los procesos internos del movimiento y administración de vacunas contribuyó en el análisis de requisitos funcionales del Centro de Salud, teniendo como resultado un sistema informático de vacunas que automatice formularios y genere reportes que se puedan manejar de forma digital.
- La metodología Scrum ayudó en la elaboración de cada módulo del sistema informático de vacunas siguiendo el plan propuesto, además que el desarrollo se lo realizó en pareja donde se mantuvo comunicación con el cliente para la respectiva retroalimentación.
- La herramienta Scriptcase permitió la construcción óptima de las aplicaciones para los diferentes módulos, ahorrando tiempo de desarrollo. Pero se encontraron también limitaciones para la solución de problemas, además de poder generar otro tipo de controles para errores.
- En la prueba de carga y estrés se pudo visualizar que el sistema informático de vacunas soporta un mínimo de 396 peticiones al servidor y llegando al valor de 400 su rendimiento empezará a responder lento y la experiencia de usuario no será la óptima.
- El sistema informático de vacunas elaborado en la herramienta de Scriptcase realiza consultas además de aplicar funciones para poder calcular y evitar operaciones erróneas e inconsistencia de datos, dichos registros se pueden visualizar en un reporte diario y consolidado mensual.
- La herramienta de Scriptcase permite al usuario visualizar y filtrar reportes registrados en la base de datos por los formularios para de esta forma poder realizar un seguimiento diario y mensual del proceso de movimiento de vacunas.

Recomendaciones

- Se recomienda respaldar la información de base de datos de manera semanal con el fin de evitar pérdida de información en caso de colapsar el sistema.
- Se recomienda revisar documentación del blog de Scriptcase para evitar errores al generar aplicaciones y revisar eventos macros que pueden ser de gran ayuda para optimizar código.
- En caso de cambiar la estructura de la base de datos y utilizar las aplicaciones generadas, se recomienda sincronizar tablas para evitar errores de campos.
- Se recomienda revisar determinadamente cada campo generado en la base de datos para así evitar errores de ingreso u operaciones al utilizar funciones.
- Se recomienda capacitar a los usuarios sobre el correcto uso del sistema, para que los beneficios que otorga sean más evidentes y duraderos.

Referencias

Bibliografía

Andreu, Ricart, & Valor. (1991). *Estrategias y Sistemas de Información*. Madrid: Mc Graw-Hill.

APR. (2021). *APR*. Obtenido de APR:

https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193#:~:text=PHP%20es%20un%20lenguaje%20de,web%20est%C3%A1

Business School. (21 de marzo de 2018). *Conexiónesan*. Obtenido de

<https://www.esan.edu.pe/apuntes-empresariales/2018/10/las-etapas-del-scrum-como-aplicar-este-metodo/>

FERRER, Y. S. (2015). *PROPUESTA PARA EL DISEÑO DE UN SISTEMA DE INFORMACION PARA EL REGISTRO DE VACUNACIÓN EN EL HOSPITAL ALBERTO SABOGAL SOLOGUREN*.

Grupo D. (21 de marzo de 2020). *APACHE HTTP SERVER PROJECT*. Obtenido de

<https://httpd.apache.org/>

Laínez, F. (2014). *Desarrollo de Software Ágil: Extreme Programming y Scrum*. IT Campus Academy.

MDN. (21 de marzo de 2021). *MDN web docs*. Obtenido de

https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript

Ministerio de Salud Pública. (2021). *Ministerio de Salud Pública*. Obtenido de

<https://www.salud.gob.ec/valores-mision-vision/#>

Salamea, C. G. (2003). *La Informática Médica y los Sistemas de Información*. Santiago.

Scriptcase blog. (2019). *ScriptCase*. Obtenido de <https://www.scriptcaseblog.net/es/scriptcase-es/scriptcase-usa-mvc/>

The PHP Group. (21 de marzo de 2021). *Php*. Obtenido de <https://www.php.net/manual/es/intro-whatcando.php>

Anexos

Anexo A: Historias de Usuario

Anexo B: Diagramas de caso de uso

Anexo C: Diagramas de secuencia.